

NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA



THESIS

RE-ENGINEERING OF A MISSION CRITICAL
SATELLITE COMMUNICATIONS COMPONENT
TD-1271B/U

by

Joe T. Hirschfelder
Laurence M. Nixon

March, 1998

Thesis Advisor:

Man-Tak Shing

Approved for public release; distribution is unlimited.

19980514 030

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE March 1998	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE RE-ENGINEERING OF A MISSION CRITICAL SATELLITE COMMUNICATIONS COMPONENT TD1271B/U		5. FUNDING NUMBERS	
6. AUTHOR(S) Joe T. Hirschfelder and Laurence M. Nixon			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) <p>Legacy software in general, and in the DoD environment in particular, presents an ever-growing maintenance challenge to program managers. The software is cumbersome, written in arcane languages and hosted on aging technology hardware. One of the options that is available to the program manager to alleviate this problem is to re-engineer the existing software product and update it to a newer language software hosted on modern equipment.</p> <p>We review existing research, select a re-engineering methodology, develop an implementation strategy and then perform a 'case study' examination of this methodology and strategy. For the case study, we take a legacy system, the Navy satellite communications multiplexer, the TD1271B/U Multiplexer, examine its existing documentation, develop a code analysis tool, perform the re-engineering on one of its sub-systems, and analyze the results. We provide observations, recommendations and conclusions on changes, enhancements and pitfalls to the methodology that will be of assistance in future re-engineering efforts of legacy systems.</p>			
14. SUBJECT TERMS Reverse Engineering, Re-Engineering, Legacy Systems, TD1271B/U.		15. NUMBER OF PAGES 176	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18 298-102

Approved for public release; distribution is unlimited.

**RE-ENGINEERING OF A MISSION CRITICAL SATELLITE
COMMUNICATIONS COMPONENT TD1271B/U**

Joe T. Hirschfelder
B.A., San Diego State University, 1979

Laurence M. Nixon
B.A., University of California at San Diego, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

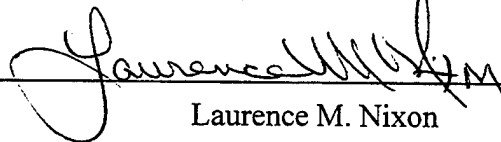
March 1998

Author:



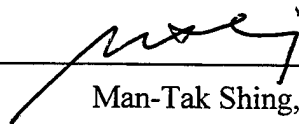
Joe T. Hirschfelder

Author:

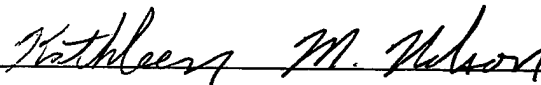


Laurence M. Nixon


Approved by:



Man-Tak Shing, Thesis Advisor



Kathleen Nelson, Second Reader



Dan Boger, Chairman, Department of Computer Science

ABSTRACT

Legacy software in general, and in the DoD environment in particular, presents an ever-growing maintenance challenge to program managers. The software is cumbersome, written in arcane languages and hosted on aging technology hardware. One of the options that is available to the program manager to alleviate this problem is to re-engineer the existing software product and update it to a newer language software hosted on modern equipment.

We review existing research, select a re-engineering methodology, develop an implementation strategy and then perform a 'case study' examination of this methodology and strategy. For the case study, we take a legacy system, the Navy satellite communications multiplexer, the TD1271B/U Multiplexer, examine its existing documentation, develop a code analysis tool, perform the re-engineering on one of its sub-systems, and analyze the results. We provide observations, recommendations and conclusions on changes, enhancements and pitfalls to the methodology that will be of assistance in future re-engineering efforts of legacy systems.

TABLE OF CONTENTS

I. INTRODUCTION.....	1
A. BACKGROUND.....	1
B. RE-ENGINEERING	3
C. OUR THESIS.....	4
II. TD1271B/U MULTIPLEXER.....	7
A. BACKGROUND.....	7
B. TIME DEVISION MULTIPLE ACCESS (TDMA)	8
C. TD1271B/U MULTIPLEXER	8
D. TD-1271B/U MULTIPLEXER DESIGN	12
III. THE RE-ENGINEERING PROCESS	15
A. PROCESS PHASES.....	15
1. <i>Phase One: Abstraction</i>	16
2. <i>Phase Two</i>	18
3. <i>Phase Three</i>	20
4. <i>Phase Four</i>	20
IV. THE CASE STUDY	21
A. THE TD1271B/U MULTIPLEXER DATA BUFFERING PROCESSOR.....	21
B. DBP INPUT/OUTPUT SIGNALS.....	21
C. THE RE-ENGINEERING STRATEGY	21
1. <i>Code review</i>	22
2. <i>Extracting the Structure</i>	23
3. <i>Extracting the Control Flow</i>	23
4. <i>Data Flow</i>	26
D. USE OF THE TD-1271 AS AN ENGINEERING GUIDE.	28
E. DOCUMENTATION REVIEW.	28
1. <i>Phase One: Architecture Definition</i>	29
2. <i>Phase Two: Requirements Definition</i>	31
F. RECOMMENDATIONS	33
V. CONCLUSIONS	35
APENDIX A BIBLIOGRAPHY	37
APPENDIX B TD-1271B/U SUBROUTINE CALL TREE.....	41
APPENDIX C PARSER.ADA	63
APPENDIX D DATA BUFFER PROCESSOR SOURCE CODE LISTING	69
APPENDIX E DATA BUFFER PROCESSOR LABEL CROSS REFERENCE.....	145
APPENDIX F PHOTOS	163
INITIAL DISTRIBUTION LIST	165

LIST OF FIGURES

Figure 2-1 Time Division Multiplexing.....	9
Figure 2-2 Typical TD1271B/U Frame Configuration.....	9
Figure 2-3 A Basic TD-1271B/U Configuration	13
Figure 3-1 The Re-Engineering And Implementation Process.....	15
Figure 4-1 Inputs and Outputs To DBP Module.....	22
Figure 4-2 DBP Control Flow Diagram.....	25
Figure 4-3 Multiple Entrances and Exits	26
Figure 4-4 IO1A Control Flow	27
Figure 4-5 DBP High Level Structure Chart	30
Figure 4-6 Control Flow Diagram	32

I. INTRODUCTION

A. BACKGROUND

One of the more complex issues associated with a software organization of significant size is the existence of large amounts of legacy software. This software typically was developed years ago, modified extensively by a variety of programmers, and is currently operating on obsolete systems hardware. The programmers that worked on the various modifications have moved on to new projects or left the organization entirely. In most instances, limited or no documentation updates have occurred. These factors all build upon each other until the software is no longer supportable. The hardware is typically old and all of the original processor reserve has been consumed, making the hardware as unsupportable as the legacy software that runs on it. The overall system is so bad that it has but one single saving grace. It works.

The program manager is charged with the responsibility of continuing to provide the services of the program while watching his ability to support the program fading. This leaves the program manager few options.

One option is to live with the system as it exists and continue to 'Band-Aid' any problems as best one can. This tends to be an expedient answer early in the life of the program, but sooner or later, this will no longer be a viable option. At some point in time the program manager will not be able to continue to support this legacy code due to physical constraints or expense.

A second option is to completely replace the existing system. This involves reviewing the current program requirements and documentation, as well as the organization operations and goals, and then to factor in those enhancements that may have forced this change in the first place. Based upon this review, a new set of requirements are generated. From these new program requirements, the program manager will need to select hardware, programming language(s), design architecture, and finally implement the software. This is then followed by a training curve to allow the operator to learn the new system. This approach has several drawbacks. The first is

usually a lack of current documentation to start the review from. It is commonplace for organizations to try to limit their maintenance costs at the expense of the quality documentation updates. Another problem are the 'hidden' enhancements; functions that are routinely use by operators, but are not documented as requirements or options even in well maintained documentation. A third is the amount of time needed to implement a full development. And, of course, there is the cost. Often for large systems the costs of a full development run hundreds of thousands of dollars, and it is not unknown for a single program's development to cost millions of dollars.

A third option is to scrap everything except for the working code and perform an in-depth analysis on the code to extract the requirements. This will tell the analyst what the code is actually doing. To this must be added the additional requirements not currently implemented. The state of the legacy code can be a drawback to this method. The success of this method is dependent upon the standards and practices used while maintaining the code. If the code was maintained by a series of programmers, each with their own standards and practices, the analyst may find extracting the information to be quite a challenge and extremely time consuming.

The fourth option is a combination of the last two, extracting the best features of each of the two options to limit the cost and time associated with updating the system. This is implemented by performing a code review and using the existing documentation as a guide to the software's basic architecture. This combination of methods reduces the amount of time required to perform the code analysis, while providing a quality requirements trace.

Each of these options has it's own advantages and drawbacks. Early in the life cycle of a software system it is fairly inexpensive to correct and fix an immediate problem. As more changes are made it becomes more expensive to correct defects or make changes. It is estimated that in a large data processing shop, 70 to 80 percent of the budget is consumed by maintenance of legacy programs [40]. Eventually the legacy system will need to be modernized.

When planning the modernization of a computerized information processing system, there are two basic items to consider: the hardware and the software that comprise the system. In the normal course of events finding a new target platform is the easier process. The speed, power, and memory capacities of computer hardware have been increasing; while hardware size and costs have been steadily falling. Replacing the software itself is another question. As the costs of hardware have fallen, the costs of software development have been rising.

This contrast of costs, cheap hardware and expensive software development, leads the program manager to look at ways of lowering the cost of the software while updating the hardware platform. The first approach would be to take the working software and port it to a new, more powerful platform. This can be done in limited circumstances, but he will normally be restricted to remaining within the existing family of processors (i.e. going from a Motorola 68030 to a Motorola 68040). In the general case, he will run into the problem of incompatibility. The existing legacy software will not run on the processor in the new platform. This places the program manager at the crossroads noted above: which of the four options is best for the project?

In this thesis we will examine the 'option four' method of modernization, that of utilizing both the existing code and documentation to develop a new baseline program hosted on a more modern hardware platform; the process of Re-Engineering.

B. RE-ENGINEERING

There are several definitions for re-engineering in the literature today [41] as described in the following:

- "Software reengineering is any activity that (1) improves one's understanding of software, or (2) prepares or improves the software itself, usually of increased maintainability, reusability, or evolvability".
- "The process of modifying the internal mechanisms of a system or program the data structures of a system or program without changing its functionality."

- "The examination and alteration of a subject system to reconstitute it in a new form and subsequent implementations of that form."
- "Re-engineering changes the underlying technology of a system without affecting the overall function, while reverse engineering is the backward engineering of a system to the specification stage. ... Re-engineering ... is the first step toward reverse engineering"

As can be seen from these quotes, the terminology of the process is not clearly defined. Some believe that Re-engineering is the beginning of the Reverse Engineering process, other that Reverse Engineering is the beginning of the Re-engineering process. For our purpose within this thesis, we will look upon reverse engineering as the systematic review and analysis of existing code to establish an information base from which to re-implement the system i.e., the first step in the re-engineering process.

Even within these somewhat conflicting terms, there are levels of re-engineering. If the only requirement is to "copy" the existing system onto a new platform, it will only be necessary to extract the architecture from the existing code and documentation before you can re-implement the same architecture in a new environment. If you want to update the architecture, then you will need to extract up to the design level before re-coding can start. Depending on the level of re-design you wish to implement you will need to extract to one level higher to allow a baseline for forward engineering at the level you want to implement.

C. OUR THESIS

For our thesis we selected a re-engineering methodology from existing research that we thought had the best application to legacy systems. We then examined this methodology of re-engineering by performing a case study. We took a legacy system, examined the existing documentation, performed the re-engineering on one of it's sub-systems and then analyzed the results. Based on this analysis and other observations we will recommend modifications to the methodology that will assist in future re-engineering

efforts. Our legacy code for this procedure is a Navy satellite communications multiplexer: the TD1271B/U Multiplexer.

II. TD1271B/U MULTIPLEXER

A. BACKGROUND

Until the space age, the high frequency (HF) spectrum was the only means of passing messages over long distances by radio. Although other frequency spectrum radios were available, their characteristics limited their use to 'line-of-sight' operations or limited data rates. With the space age came the ability to place an Ultra High Frequency (UHF) radio repeater in a geosynchronous orbit. By placing a transmitting station high above the earth, it's 'line-of-sight' earth coverage was greatly expanded. The characteristics of the UHF Satellite Communications lended itself well to the needs of the Navy. It provided dependable over-the-horizon communications capability at accelerated data rates. The high data rates allow the transfer of data and voice. As the capabilities became more developed and terminals became more prolific, time lags and data delays became more regular. System throughput began to suffer. This left the Navy with a limited number of options. One was to reduce the number of users and the amount of data they were sending. This was an extremely unpopular option to the Fleet. They had grown to depend on the types and levels of data being supplied, and their effectiveness would suffer if any of the data were discontinued. The second option was to acquire more UHF channels. This was not a viable option either. The commercial world was rapidly expanding into the UHF spectrum, and lobbying congress to decrease the number of channels reserved for military use and sell them to commercial interests. The only option left was to find ways to utilize the existing channels more efficiently. To meet the need for more efficient usage of the UHF channels the Navy developed the TD-1271B/U Multiplexer.

The basic philosophy behind the TD1271B/U Multiplexer, or MUX, is to use Timed Division Multiple Access (TDMA) principals to integrated multiple nets, each operating on an independent channel, onto a single channel. By multiplexing several channels onto a single channel more data transmission path options become available.

B. TIME DEVISION MULTIPLE ACCESS (TDMA)

In TDMA, data streams at moderate to slow data rates are input via baseband data ports and buffered. At a point in time, predefined for each data port relative to a time marker, this data is transmitted out at a data rate that is significantly faster than the input data rates. While one baseband port's data is being burst out to the satellite, the other ports data streams are being buffered, awaiting their burst opportunity. Using appropriate bursting speeds, none of the baseband units will suffer any loss of continuity in their data. Each will transmit and receive data just as if on a dedicated channel, without realizing the data was multiplexed. This concept is illustrated in Figure 2-1. The key to this process is being able to transmit at a faster data rate than all of the baseband ports data rates combined. The elapse time between the start of one timing mark and the beginning of the next timing mark is generally referred to as a frame. The period of time allocated for a transmission burst is often referred to as a time slot. Depending on the particular protocol for the system, the TDMA Frame may contain a series of time slots with equal time periods or may have time slots with varying time periods.

C. TD1271B/U MULTIPLEXER

In the TD1271B/U Multiplexer (MUX) operating in the TDMA-1 25 kHz mode, a frame is defined as a period of time equaling 1.38666 seconds.¹ This frame not only has varying time slot periods, but is re-configurable based on users needs. The allocation of time slots is determined by the definition of a three digit (hex) Frame Format code. Each of the digits defines the configuration of one of the three user segments contained in the frame. A typical MUX frame is diagrammed in Figure 2-2.

The MUX waveform accommodates multiple input/output (I/O) bit rates and radio frequency (RF) symbol burst rates, as well as dynamic assignment of users between channels. Operation of the MUX waveform is transparent to the user baseband equipment, except for TDMA frame-time delays.

¹ The TD1271 MUX is also capable of a secondary waveform (TDMA-2 or TDMA-3) that utilizes a 5kHz channel. Discussion of this 5 kHz waveform is beyond the scope of this thesis.

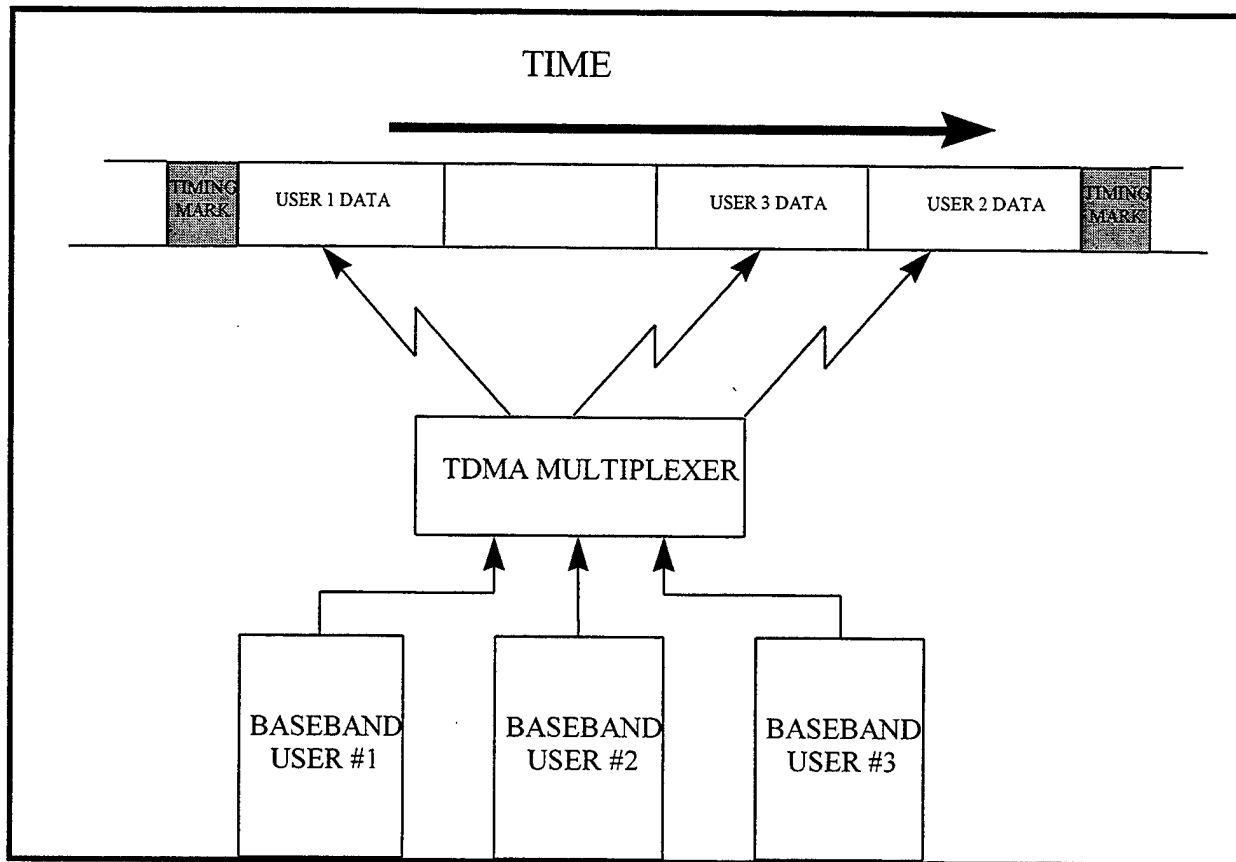


Figure 2-1
Time Division Multiplexing

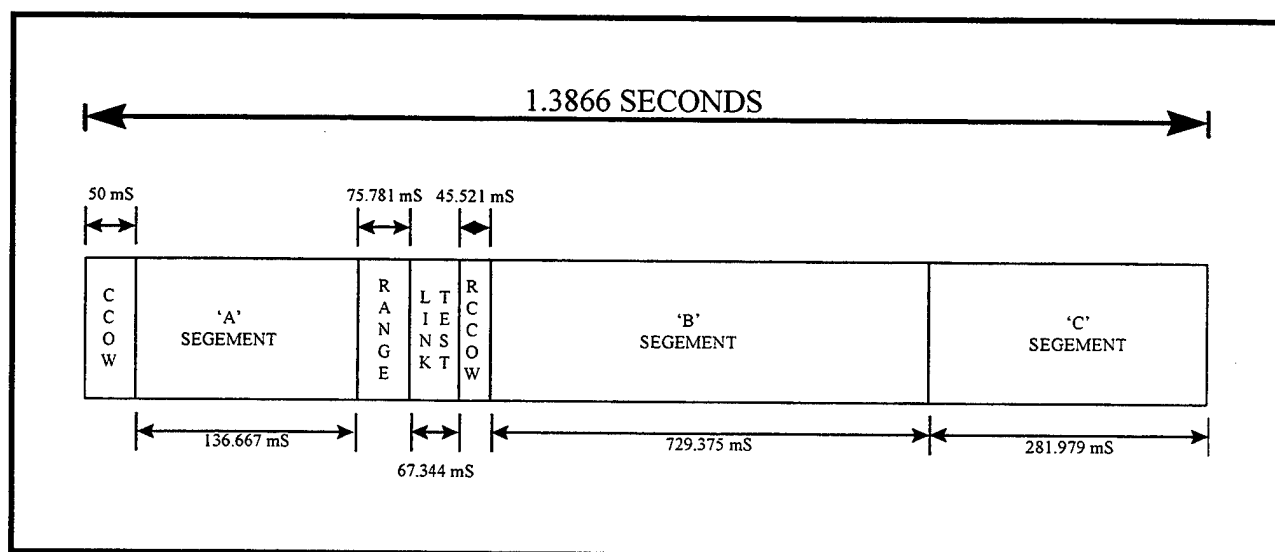


Figure 2-2
Typical TD1271B/U Frame Configuration

The MUX waveform is organized as a repetitive frame structure with 1.386-second frames, as depicted in Figure 2-2 (Frame Type 2 is depicted²). Frames are subdivided into slots assigned for: (a) orderwire communications [channel control orderwire (CCOW) and return channel control orderwire (RCCOW)]; (b) system support functions (range time slot and link-test time slot); and (c) user segments A, B, and C. User segments A, B, and C comprise multiple user slots. Each of the 5 slot types in the MUX waveform is identified below:

1. CCOW slot. The CCOW slot is used by the channel controller to broadcast CCOWs. One of the 20 defined CCOWs is transmitted by the Channel Controller in each frame. The CCOW contains the information required to control operation of the waveform and provides a timing mark for the frame.
2. RCCOW slot. The RCCOW slot provides users the opportunity to send data, including requests for access, to the channel controller.
3. Range slot. The range slot provides time for users (including the channel controller) to transmit and receive a data burst for the purpose of calculating the range to the satellite, based on measured round-trip delay time.
4. Link test slot. The link test slot provides time for users (including the channel controller) to transmit and receive a test data stream to determine the current operating conditions, based on the measured bit error ratio (BER). The link test slot is also used to perform dedicated range measurements, when the link test slot is in an even-numbered receive frame. This is known as an even link test (ELT) slot.

² In order to support half duplex terminals, the MUX waveform supports a Frame Type 1 that moves the 'C' segment slots between two halves of the 'B' segment. The half duplex network would transmit in the first 'B' segment and receive in the second 'B' segment.

5. User slot. User slots are allocated for transferring data between users. Associated with each user slot is a specific baseband data rate, encoding schema, and burst rate.

Each slot within the MUX waveform consists of three elements: (1) a preamble for receiver carrier, bit, and data synchronization; (2) data; and (3) guard time to avoid user-to-user interference. The receiving equipment achieves carrier and bit synchronization for each slot in the waveform.

The MUX waveform is designed to operate on 25-kHz non-processed channels of current and planned UHF SATCOM satellites.

The MUX accepts input from digital communications systems via its four baseband data ports. The baseband data ports will accept data at rates of 75, 300, 600, 1200, 2400, 4800, and 16000 bits per second. In order to maintain data integrity, Forward Error Correction (FEC) encoding is used. This FEC coding is convolutional, with interleaving. Encoding rates of $\frac{1}{2}$ or $\frac{3}{4}$ are used. The MUX will output the data to the radio for transmission at one of three burst rates; 9,600, 19,200 or 32,000 symbols per second. The interface between the radios and the MUX is over 70 MHz intermediate frequency carrier lines. Reception of data is processed in a similar, but reversed, fashion.

One of the most critical functions of the MUX in this process is to maintain consistent timing within the frame. This timing begins with the reception of the timing mark to begin the frame. In the MUX system, all timing is based on the reception of the Channel Control OrderWire (CCOW). The primary CCOW is the Master Frame CCOW and contains the current frame format number, frame sequence number, and cryptographic data needed to define operations on this channel. The Master Frame is transmitted every eight frames. Other CCOWs are available to direct net operations, request data from the various terminals³ operating on the net and maintain timing.

³ Although we are only concerned with software and hardware of the TD1271B/U, there are other terminals that are operating on nets with the TD1271B/U. When discussing operations in general, we will use the term terminal to include all of the possible systems operating on a channel.

CCOWs are generated and broadcast by a specifically designated terminal. Each channel will only have a single terminal generating CCOWs at a time (the Primary Channel Controller (PCC)) although other stations could be designated as alternate controllers (ACC).

Once the MUX has successfully received the CCOW, the next critical event is to calculate the terminals distance to the satellite in units of time. The standard time unit in the MUX is the 'time chip'. There are 19,200 time chips in one second. In order for all terminals on the net to receive data in the proper locations within the frame, all transmissions are timed to be correctly placed within the frame at the satellite. In order to insure this 'perfect' frame at the satellite, and by extension at each receiving terminal, the distance to the satellite must be known. This is achieved first at power-up by sending a ranging burst during the designated ranging slot, and updated periodically using the Even Link Test (ELT) slots.

D. TD-1271B/U MULTIPLEXER DESIGN

The basic TD-1271B/U installation comprises a TD-1271B/U multiplexer, a KGV-11 encryption device, a radio transceiver (a Navy WSC-3 pair is typical) and from one to four baseband user devices (ANDVT, ON-143 or teletypewriters would be typical). Figure 2-3 provides a block diagram of a basic TD-1271B/U's configuration.

The software in the MUX is divided into three Computer Software Configuration Items (CSCI); the System Control Processor (SCP), the System Timing Processor(STP) and the Data Buffer Processor (DBP).

Each CSCI has a Memory Program Unit (MPU), an interface board, and at least one 24 k byte Ultra Violet Erasable/Programmable Read Only Memory (UVEPROM) board that contains the executable program. The SCP CSCI utilizes two UVEPROM boards.

The Memory Program Unit (MPU), hosts the Central Processor Unit (CPU), a Motorola 6800, along with the System Random Access Memory.

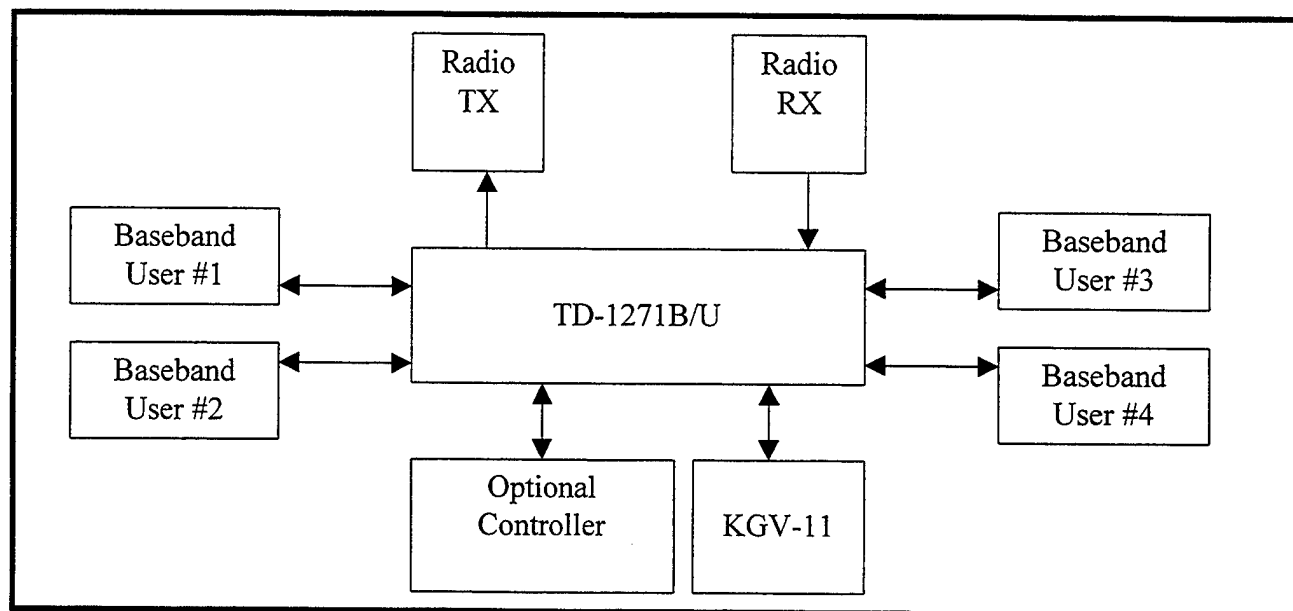


Figure 2-3
A Basic TD-1271B/U Configuration

III. THE RE-ENGINEERING PROCESS

A. PROCESS PHASES

Re-engineering begins with the assembly of all resources of a software system. This includes, but is not limited to the software source code, design documents, software development folders, test reports, software trouble reports, sample inputs, sample outputs, and, if practical, interviews of the implementers and/or maintainers. Typically the only reliable source of information is the software source code and for that reason re-engineering of an existing software system starts at the code level. The software documentation, many times, are not kept current with the software; however, it can still be a useful source of information and can be used as an engineering guide. The process in which existing software is transformed into a new implementation can be described in the Figure 3-1 [30].

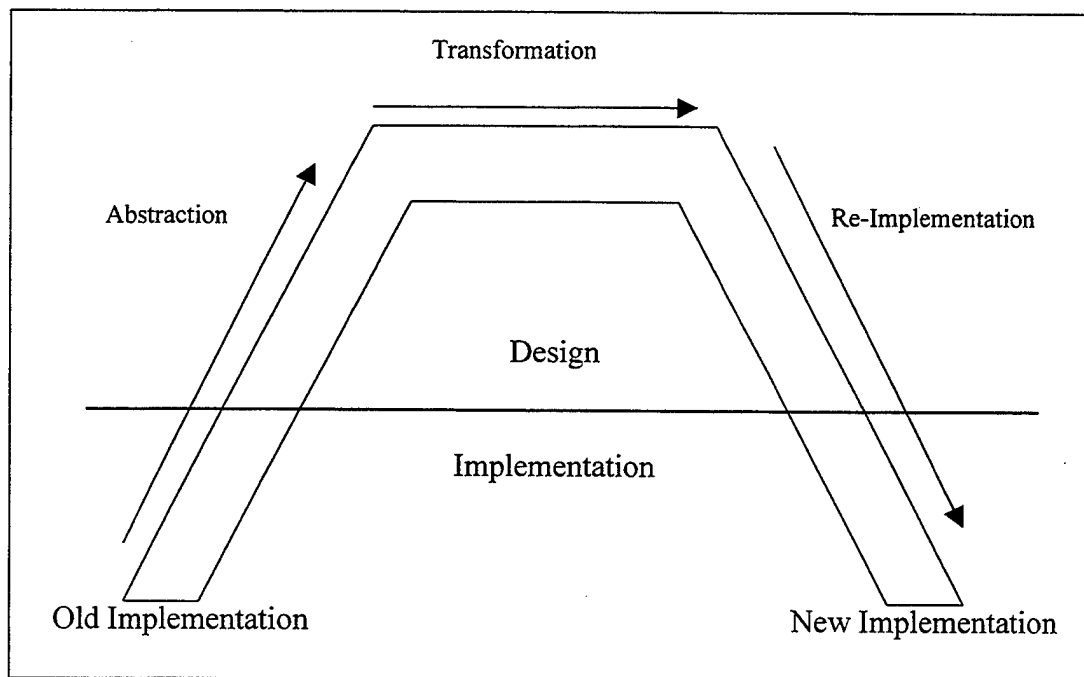


Figure 3-1
The Re-Engineering And Implementation Process

There are three phases to the re-engineering process. The first phase, abstraction, extracts the detailed design information from the documentation and the code. For those familiar with MIL-STD-498, this is analogous to creating a Software Design Document from the existing system. The second phase allows for the modification of the design by adding, deleting, or modifying requirements. The third phase is a re-implementation of the system. The focus of this thesis is the reverse engineering process that recovers the design from the old implementation. This methodology will be validated by executing the re-implementation of the recovered design using a different language and different processor than the old implementation.

1. Phase One: Abstraction

In this phase, the documentation and code can be used to provide the information needed to support the re-creation of the system design. Since the TD-1271 documentation has been shown to be unreliable, the source code will take precedence in this phase with the documentation providing background information. The products developed during abstraction are as follows [30]:

1. **Structure Chart.** This is a hierarchy of the modules as extracted from the program where each node in the chart corresponds to a procedure or function in the program. Since the goal of abstraction is to remove the language dependencies from the code, it is important to treat library calls as part of the system and include them in the structure chart. The language selected for the re-implementation phase may not provide the same library services. Any language dependent library calls may need to be manually re-implemented. The structure charts for this thesis are based upon the structure charts identified in reference [37]. The structure charts provide a visual representation of the software system that is useful in determining organization and dependencies. This information, once derived, is used to develop the structure charts for the enhanced system.

2. Module Description. This description is a combination of the module's inputs, processing, and outputs, with the algorithms and constraints identified. This description is produced for each module in the structure chart. In addition to a narrative of the module's processing, a language independent form using a specification language such as SPEC [42] or a Program Design Language (PDL) is used to capture the requirements of the module. For this documentation to be useful it is important to verify, through code reading, that the supporting documentation is up-to-date. A case tool that allows this information to be tagged to each element in a structure chart would be very helpful. One of the original goals was to identify a tool to preserve this information as part of this thesis, but due to budgetary considerations no tool was procured.

3. Data Flow Diagram. At a module level the data flow diagram shows the flow of data between the modules. Within the module that the data flow diagram is decomposed, showing at each successive decomposition level the transformations that occur. This information is also used to develop the algorithms that need to be documented when the design is re-implemented. These algorithms can be described using mathematical notation, a Program Design Language, or other notation. These algorithm descriptions are added to the module description in step 2. The process consists of constructing the DFD of level $i+1$ from level i . Each level explodes the data into transforms, data repositories, and data flows [37]. Data repositories will be represented as boxes, data transforms as circles, and data flows as connecting lines between transforms and data repositories. The transforms are then used in phase 2 to develop the detailed design.

4. Control Flow Diagram. This will identify the control structure of the program between the modules. This information is used to identify the logical relationships between the modules, verses the physical relationship described in step 1. [39] will be used to create this diagram.

5. State Transition Diagram. This will identify the states and modes in which the software can reside and identify the events that cause a change in the state of the software and the processing that is to occur due to the state change, and will identify the modules active for each state. There may be more than one state transition diagram depending upon the concurrency of the system. The notation for this diagram will follow those in [39].

Once the design information has been extracted, the resulting recovered design is reviewed for consistency. Any missing items of information should be investigated and added to the recovered design [30]. While it is the goal of this first phase to extract the design and leave behind the vestiges of the language and platform, the design may still have an organizational structure that was due to the implementation language and platform. Before entering phase two, the extracted design information can be modified to refine the design for portability. For example, the TD1271B/U uses multiprocessors with serial communications between them to perform the processing required. When the system is re-engineered, it is anticipated that this system will be implemented using Ada tasking on a single microprocessor. The detailed design should be abstract enough to make this possible.

2. Phase Two

Phase one used the supporting documents and the code to extract the detailed design information. This phase attempts to link the detailed design information with the requirements through the use of a requirements traceability table. This table binds the modules with the requirements that they partially or fully implement. This table will need to support one to many and many to one. That is, one requirement may be spread out over one or more modules and many requirements may be identified with one module. Depending upon the state of the supporting documentation this can be an easy task or a major effort. Requirements are needed if the system being reengineered is also having the requirements modified as well. Once the requirements have been modified, the modules affected need to be updated. It is not unusual at this phase to add and delete

modules from phase one. This phase is identified for completeness only; this thesis will not attempt to construct a requirements traceability table.

The TD1271B/U was developed under DOD-STD-1679 and any re-engineering of the TD1271B/U would use a tailored MIL-STD-498 approach. The purpose of the re-engineering effort is to develop, using the existing resources of the project, those products necessary to pass the exit criteria of the detailed design phase and allow the system to be coded. The primary product that is developed during the detailed design phase is the Software Design Document. When looking at the data item description (DID) the following sections are used to provide the programmer with the information necessary to code the system. From the re-engineering activities performed on the TD1271B/U in the three phases described above, the following module information will be developed:

- Module Name
- Module Description
- Constraints, Limitations, or Unusual Features in the Design
- Special or Unique Algorithm Description
- Called by
- Calls
- Triggering Event (tasks only)
- Implicit Inputs
- Explicit Inputs
- Implicit Outputs
- Explicit Outputs
- Hardware Dependencies
- Sizing and Timing
- Informal Test Procedures

If computer aided design tools were used during phase two, such as the Computer-Aided Prototype System (CAPS), much of the information needed for the SDD is contained within the various reports that the tools produce.

3. Phase Three

At the completion of phase two the design of the current system has been extracted. Before starting the re-implementation of the system for the new target language and hardware, any changes to the design is done in this phase. This could be a re-organization to take advantage of any special capabilities of the compiler or hardware. In addition, this would be the time to modify the requirements of the system and fold those changes into the design. Upon completion of this phase the design should reflect the system as wanted by the customer.

4. Phase Four

The output of phase three is used in this phase. This is the coding of the extracted and possibly modified design. The detail design is re-implemented using the desired methodology (i.e. OOD, structured design, etc.) in the desired language for the desired target platform. This phase is executed in the same manner as any new development.

IV. THE CASE STUDY

A. THE TD1271B/U MULTIPLEXER DATA BUFFERING PROCESSOR

The Data Buffer Processor (DBP) provides the elastic buffering required to pass continuous data between the Systems Control Processor (SCP) and I/O ports while transmitting or receiving data in bursts over the satellite channel. The heart of the DBP consists of 24K bytes of EPROM containing the software program. The program is a loop structure which sequentially scans internal control points and the baseband users to determine the actions is required. If, for example, a user terminal has entered a transmit request at the baseband I/O port, a check is made of the port requirements, the port is configured to receive data and the transmit acknowledge is sent to the baseband device[43]. Figure 4-1 provides a block diagram of the relevant input and output signals to the DBP.⁴

B. DBP INPUT/OUTPUT SIGNALS

The DBP[49] data flow is described in Figure 4-1. Figure 4-1 is a description of data and control signals that are input to or output from the DBP[49].

C. THE RE-ENGINEERING STRATEGY

One of the primary goals of this re-engineering effort was an attempt to use the existing source code to extract the design of the system and to produce well designed, portable code. Ada was used as the target language during the extraction process. Since this software was originally written before the advent of structured design, we didn't attempt to use an Object-Oriented Design approach, but used a structured design approach.

⁴ This listing is not complete for the TD1271. Certain hardware control lines are system dependent and would not be applicable to this re-engineering effort.

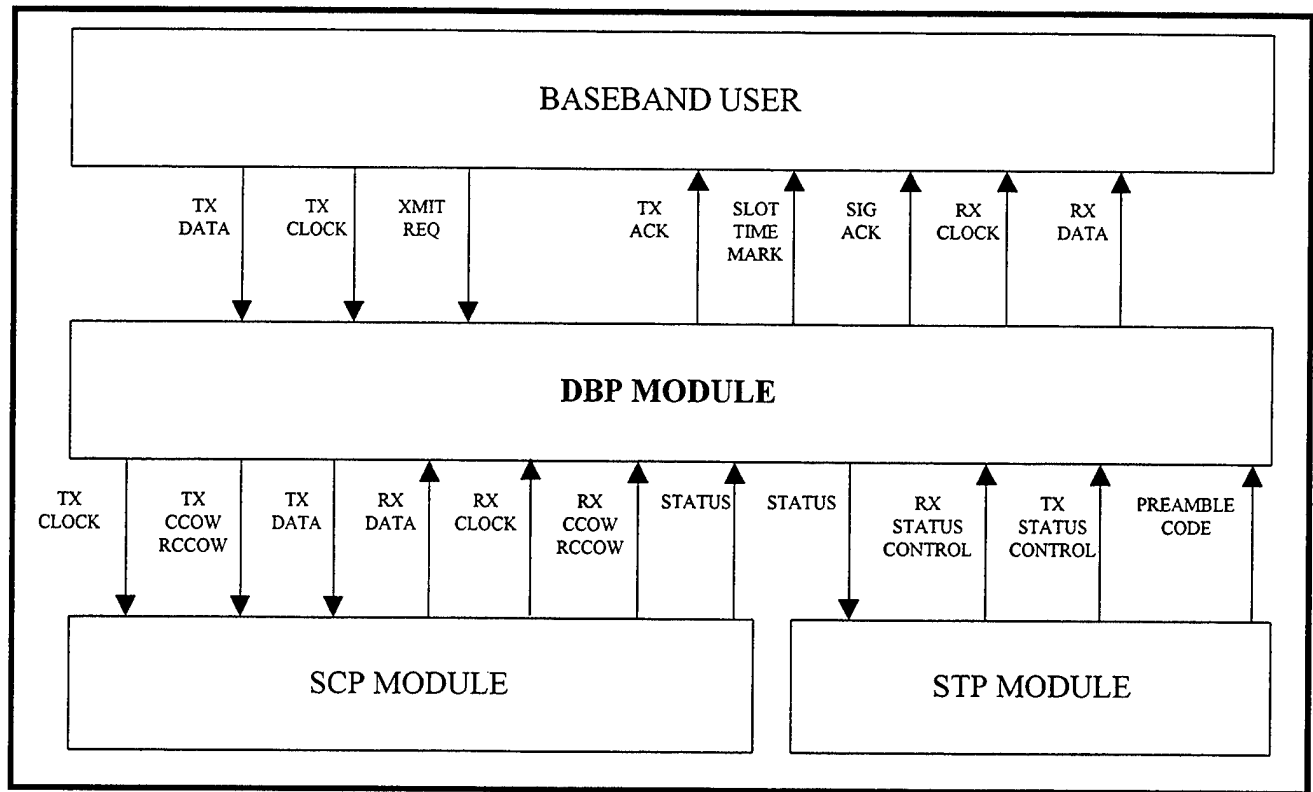


Figure 4-1
Inputs and Outputs To DBP Module

As outlined in Chapter 3, the first phase of the re-engineering effort is to develop Structural Diagrams, Module Descriptions, Data Flow Diagrams, Control Flow Diagrams, and State Transition Diagrams. The code was used to extract the structure, control flow, and state transition diagrams. The documentation was used to develop the module descriptions, provide a high level data-flow diagram and to verify the control flow and state transition diagrams whenever possible.

1. Code review

As noted by T. Bigerstaff [10], a key objective of design recovery is to develop structures that will enable a software engineer to understand the program. Our first step in this process was to identify and partition the code into procedures. This proved to be a task of greater difficulty that we had anticipated. Not only were we dealing with a

somewhat cryptic language, we were also faced with a coding style that extensively utilized 'jump' and 'branch' commands to navigate from one set of instructions to another. This programming style was intended to reduce the total size of the code by reusing existing code rather than re-writing it. Programmers coding during this period of time were highly praised if they could reduce the length of the overall program by even a few bytes, regardless of the long term maintenance problems created.

2. Extracting the Structure

The first step was to extract the modules from the code. Modules were defined as a section of code referenced in a Jump to Subroutine (JSR) or Branch to Subroutine (BSR). Even though it was recognized early on that this effort would focus only on part of the TD-1271 system it was important to extract the structure of the entire system to identify the areas in which the rest of the re-engineering effort would focus.

The code for the three major sub-systems were examined and mapped using a Microsoft Excel spreadsheet to record the results (see Appendix B - Subroutine Call Tree). It should be noted that this Subroutine Call Tree is not complete. The TD-1271 is not a procedural based design, that is it makes extensive use of GOTO statements for control between subroutines. The effort to extract the subroutine calls using the JSR/BSR opcodes was sufficient to identify which sub-system to apply the more rigorous code examination to. To produce the Subroutine Call Tree each file comprising the TD-1271 was opened in a word processor and a search was conducted on the words JSR and BSR. At each occurrence the subroutine containing the call and the subroutine that was being called were entered into the Subroutine Call Tree. Duplicate entries, the same subroutine call from the same subroutine, were not noted. The Subroutine Call Tree contains 1183 subroutine calls. After examining the results of this effort the Data Buffer Processor was selected for further investigation.

3. Extracting the Control Flow

In order to extract the control flow from the code it would be necessary to have some sort of cross reference available to consult. There weren't any tools available to do this so one was developed. A parser was written in Ada using an Alsys host compiler for

the IBM-PC. This parser uses two passes to build a cross reference for the labels in the source code and a source code listing with line numbers. The first pass identifies each label and the line of code where it is defined. The second pass identifies the line number of code where each reference occurs. References to labels that are not found are reported as errors. The only errors encountered were relative indexes based upon a label (e.g. LABEL1+2). The source code listing is in Appendix D and the Cross Reference Listing is in Appendix E. There will be references to line numbers in the source code from Appendix D later on.

A detailed examination of the DBP code identified the inter-subroutine control flow and a disturbing software design methodology, or to be more accurate - lack of one. As you can see by the control flow diagram, Figure 4-2, the designers used some programming practices that made it very difficult to extract the control flow. The first difficulty encountered was the use of multiple return from subroutine (RTS). Rather than having one entrance and exit from the subroutine there were multiple exits. This was further complicated by the use of GOTO statements where a JSR was used to enter a subroutine and both a RTS and a GOTO statement was used to return (IO1PT to IO1LF in Figure 4-2). There were subroutine calls where the called subroutine executed a GOTO statement to pass control, not back to the caller, but to another routine (IO1PT - IO1LS - IO2PT in Figure 4-2). There were manipulations of the return stack to short circuit the return from a subroutine call (Appendix D, line numbers 1044-1045 and 1099-1100 in IO1PT and repeated for the other three channels). There are examples in which a routine has multiple entrances and exits (Figure 4-3). The way this code was designed made it extremely difficult to convert to Ada, and impossible to meet the goal of having code that is well designed following this conversion. The result was becoming as poorly designed as the original. As stated earlier, this software was designed prior to the introduction of what is now considered modern programming standards. Design decisions were made based more upon memory usage and execution speed than having a system that was easy to maintain.

The subroutine IO1PT was examined for developing the inter-subroutine control flow. This was based upon the conditional statements within the code. This inter-

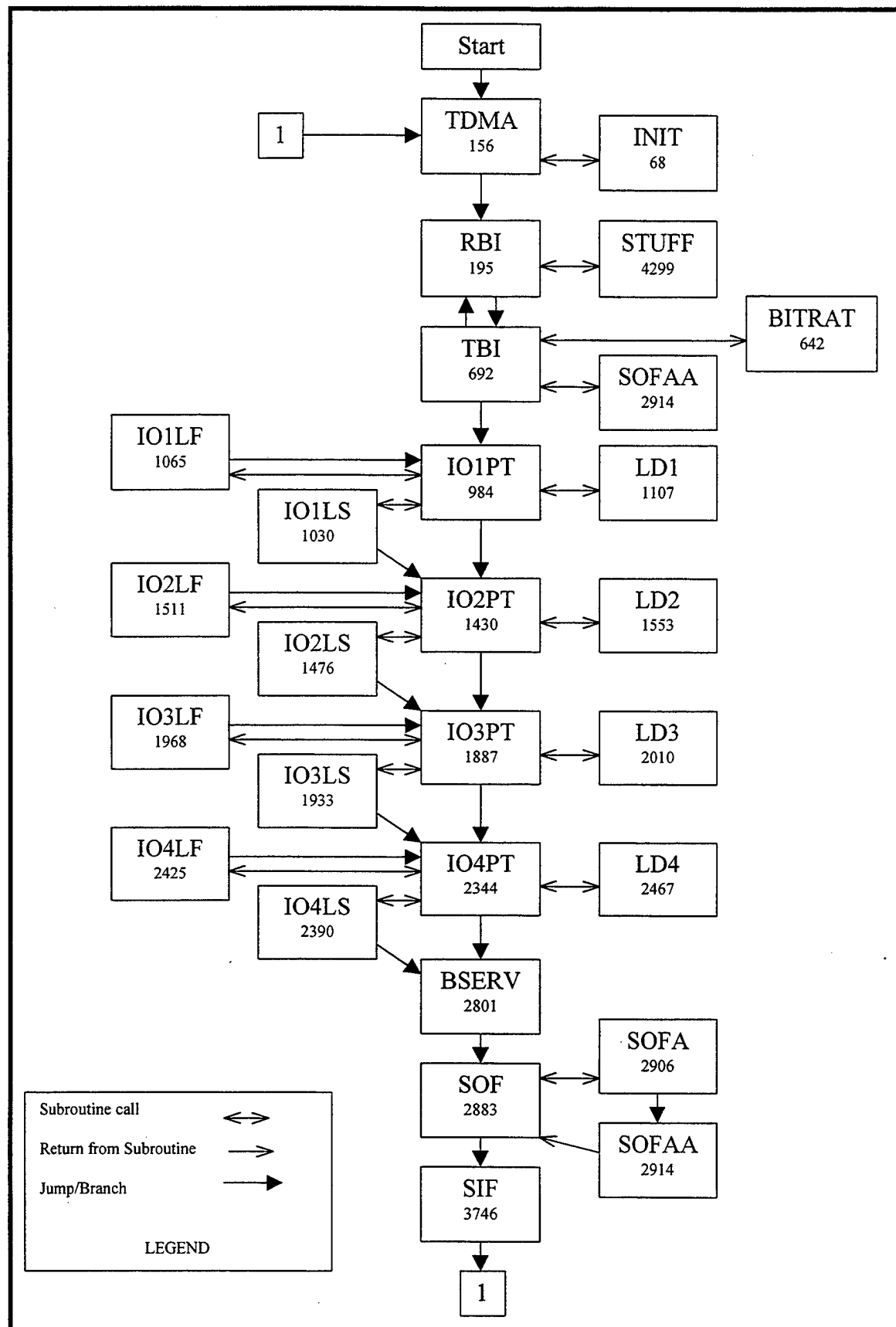


Figure 4-2
DBP Control Flow Diagram

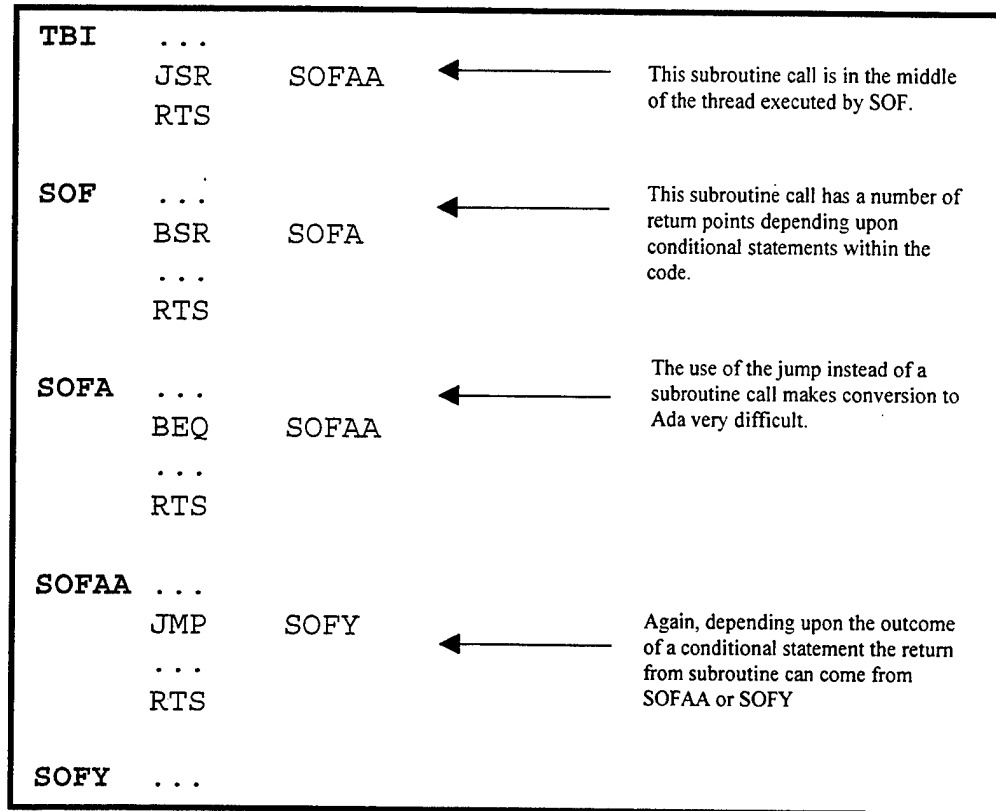


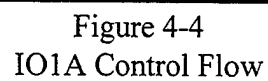
Figure 4-3

Multiple Entrances and Exits

subroutine control flow was found to be as difficult to convert as the intra-subroutine control flow. The control flow for subroutine IO1A can be visualized with Figure 4-4. As Figure 4-4 depicts, this would be an example of "spaghetti code". In order to overcome this spaghetti code problem it became necessary to reduce the code to 'mini' routines, adding 'jumps' to the end of each code fragment, so each fragment would be a stand alone routine. Each fragment could then be moved at will to try to untangle the subroutine's control flow. It too was far more difficult than it was worth. The result was a subroutine that was poorly designed.

4. Data Flow

The data was analyzed to attempt to produce a data flow diagram. The primary data flow used ringed buffers. The declared buffer started at its own starting address and ended at the starting address of the next declared buffer (e.g. an index pointer was set to



the starting address of a buffer and was incremented until it equaled the starting address of the next buffer). This hard coding is the easiest to overcome, but it is a glaring example of the way the software was written. These ringed buffers can be easily defined using a ring buffer generic package in Ada where the data is inserted and extracted using procedure and function calls. Error handling can be included to cover the cases of underflow and overflow. The size of each buffer is then easily established during elaboration based upon the size specified. It was determined that the best source of information to establish a high level data flow diagram was the documentation. This data flow is shown in Figure 4-1. The data flow between the modules of the DBP are primarily indirect by using data stores. It was decided that the effort to describe this data flow was of little use and was not done.

D. USE OF THE TD-1271 AS AN ENGINEERING GUIDE

It was during the effort to establish the control and data flow diagrams that the determination was made that the TD-1271 should be redesigned from the top down. The source code as designed didn't support the re-engineering effort except as an engineering guide during the redesign processes.

The interrupt handlers can be used to establish the tasks responsible for the input and output of data to the DBP subsystem. While the modules are not suitable for re-engineering, the organization of the main modules can be useful in partitioning the new system. You can see by Figure 4-2 that there are four channels that are serviced in a round robin method. The new system can also use this organization. The most valuable asset the current TD-1271 has is by providing a system that works. A comparison between the behavior of the system to various inputs and what is specified in the documentation can give a good indication on the state of the documentation. It provides a strong baseline between stimulus and response that can assist in the formation of the new design.

E. DOCUMENTATION REVIEW

As was noted in the previous section, attempts to extract the requirements directly from the code became a prohibitively tedious and difficult task. This left only the

original documentation with which to attempt to baseline the requirements. The primary documentation available for the Data Buffer Processor program includes the Program Performance Specification (PPS) [49], the Program Design Specification (PDS) [50]. Also available are the system level PPS [46] and PDS [47] for the TD1271B/U. The factors and problems noted about documentation in Chapter I are evident in the TD1271B/U's documentation, and bring it's overall accuracy into question. The documents do however provide the basic requirements as they were known as late as 1987. It is these requirements that we shall briefly examine within this section relative to the re-engineering plan detailed in Chapter III.

1. **Phase One: Architecture Definition.**

- a. *Structure Chart*

The DBP PDS [50, page 3-1] provides a basic structure chart. This is reproduced here as Figure 4-5. This structure chart does not include the modules for initialization and setup.

- b. *Module Description*

Module descriptions for the six 'run time' modules are provided in both the PPS and in the PDS for the DBP. Setup and initialization are not specifically listed. Briefly these six modules are:

1. Receive Burst Initialization Module (RBI). This function processes the receive burst notification from the System Timing Processor. It initializes the DBP's hardware, the I/O ports flags and pointers, and the Service Receive Burst FIFO function.
2. Transmit Burst Initialization Module (TBI). This function processes the receive burst notification from the System Timing Processor. It initializes the DBP's hardware, the I/O ports flags and pointers, and the Service Transmit Burst FIFO function.

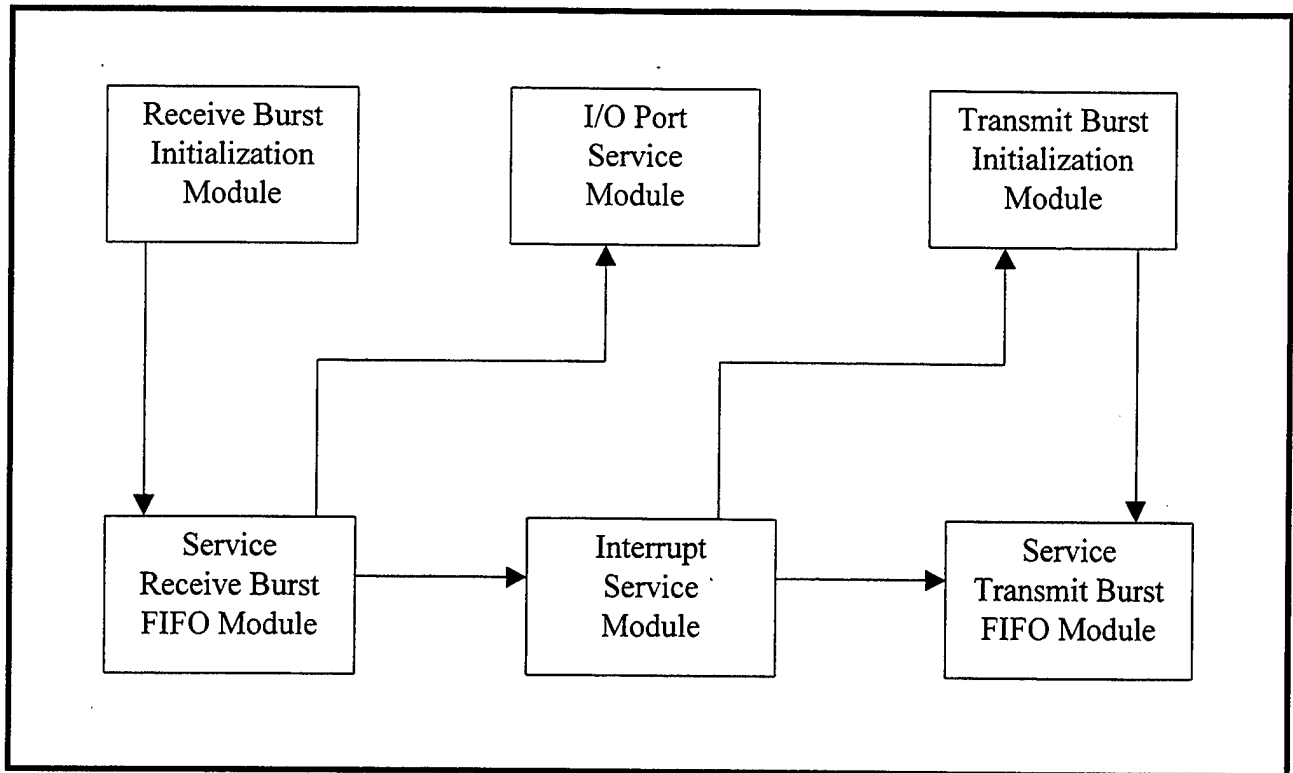


Figure 4-5
DBP High Level Structure Chart

3. I/O Port Service Module (IO1PT-IO4PT). These four modules services each of the four baseband ports. If the I/O Transmit Request input is true, the DBP will accept I/O port data and store it in the port's data buffer. If the I/O Transmit Request input is false and there is receive burst data for the port, then the I/O port service will output this data from the data buffer to the baseband equipment.

4. Service Receive Burst FIFO Module (SIF). This module inputs receive burst channel data and stores it in the data buffer.

5. Service Transmit Burst FIFO Module (SOF). This module outputs the I/O port's transmit data from the data buffer to the burst channel.

6. Interrupt Service Module (IRS). This module communicates with the System Control Processor. It receives I/O port bit rate and connection status, and it receives/transfers orderwire messages on an interrupt basis [49].

c. Data flow

The Data Flow is somewhat more difficult to extract from the documentation. By definition in the DOD-STD-1679A [51], the full definition of data variables would be documented either in the Program Design Document or in the Database Design Document. To date we have not found any reference to either of these documents. It is our belief that they were tailored out of the documentation package for the original development. This leaves some of the inputs and outputs listed in the DBP PPS and PDS open to question as to their origin and content. Clearly this is an area where it would be necessary to track the variables through out the code to ascertain their design and functions. In addition to the program variables, there are an assortment of Key Lines, Clocks, and Interrupts that need to be processed and input. Figure 4-1 provides a block diagram of the inputs and output from the DBP module.

d. Control flow

From the DBP PDS we were able to extract the basic control flow diagram. This diagram (Figure 4.6) would provide the sequencing of tasks in the new software module. It would also provide the basis for the modules for the Data Flow diagram. In addition to this 'basic' control flow, there is also an Interrupt handling module capable of stopping current processing and shifting control to another module.

2. Phase Two: Requirements Definition.

The second part of the re-engineering effort is to utilize the data gathered from the first phase to develop requirements. The high level requirements can be easily extracted from the module descriptions.

- R1: Will receive burst timing from System Timing Processor.
- R2: Will initialize hardware and software flags and statuses
- R3: Will input data from the Baseband ports and store.
- R4: Will receive data from the channel and store.
- R5: Will forward input data from storage to the channel.

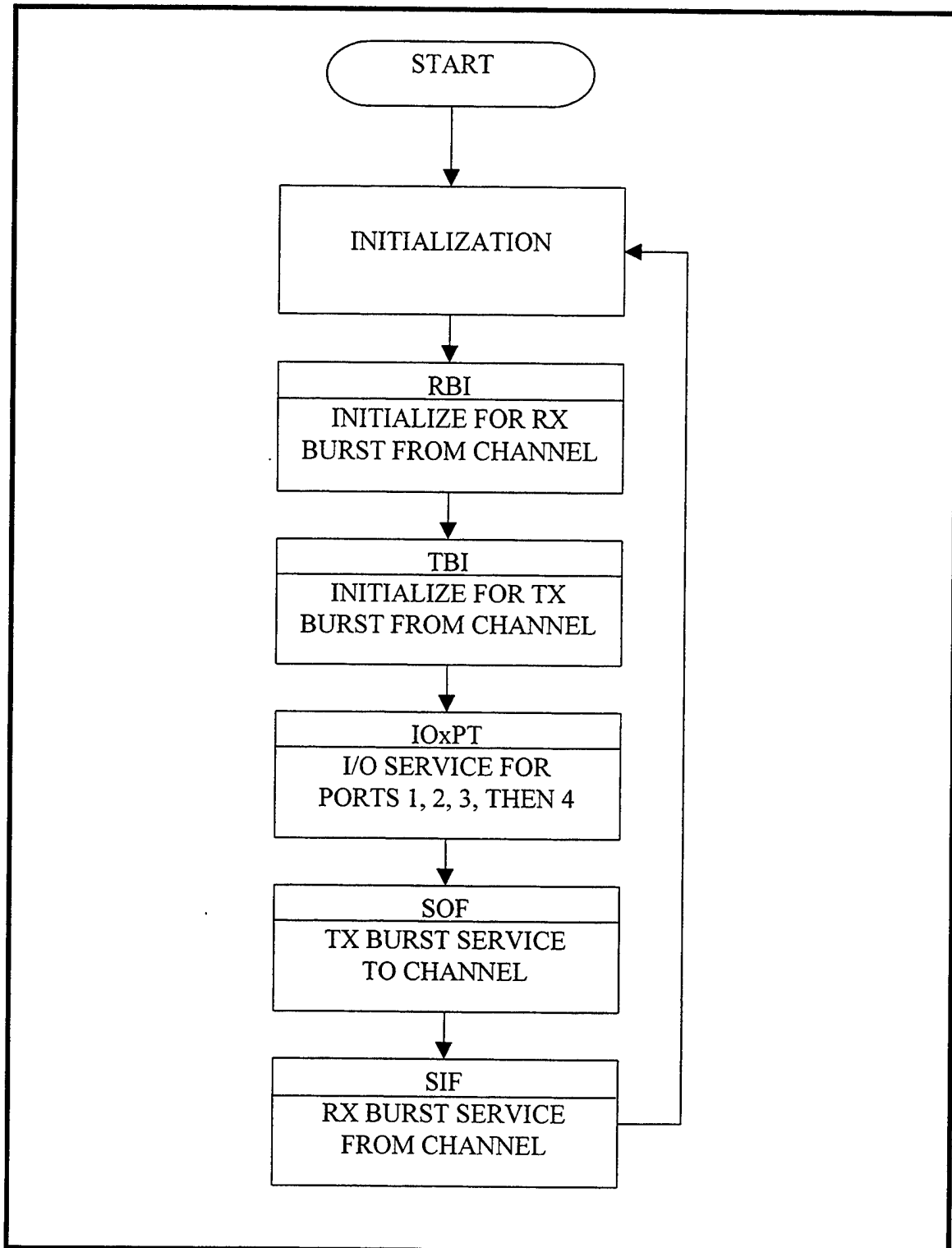


Figure 4-6
Control Flow Diagram

R6: Will output receive data from storage to the Baseband port.

R7: Will receive and transfer orderwires to and from the System Control Processor.

R8: Will receive port status and connection information from the System Control Processor.

From this point, development of more detailed requirements would progress along normal development lines. The advantage that would be gained in this area would be the ability to check the foundation of a requirement in the code. In this way the 'forward' engineering effort would have a existing system to validate the responses against.

F. RECOMMENDATIONS

Effort can be expended to convert the TD-1271 to a high order language and have it run on any number of modern day processors. To do so without reengineering the system from the top down will in effect doom this system to the same fate it has fallen to today: unstructured and not maintainable.

One of the obvious recommendations would be to use a modern programming language. Because of the use of ring buffers between the subsystems, a method that is valid in a new design, Ada would be the best choice. This would allow the use of generic packages to handle the buffers. Ada's tasking and the ability to tie the task's entry point to a hardware interrupt would be another valuable feature that makes Ada the first choice.

The original thesis called for timing studies to be conducted on the re-engineered code. Since the code never reached the state where these studies could be performed there aren't any hard numbers to address. However, the current system runs on three Motorola 6800 microprocessors. The increase in the processors throughput since the Motorola 6800 introduction would give a good indication that the entire system could fit on any number of modern day microprocessors. The hard real time requirement is handling the input from the modem. This data is transmitted at a maximum rate of 16K bits/second. Most real-time embedded applications use Wind River's VxWorks operating system. This operating system allows the application to read a serial port with a variable input size. The programmer specifies the file descriptor for the serial port, the address in

which the data is to be read, and the number of bytes to read. In addition, VxWorks provides an optimized task switch for interrupt handlers that has the overhead of a function call. Once the data has been read in the data is output as processed. By proper scaling of the input data from the modem, the processing percentage spent on handling the interrupt can be less than 10%. With these considerations our recommendation would be either the Motorola 68040 or the Motorola Power-PC.

V. CONCLUSIONS

This thesis was originally planned to be a fairly straight forward application and review of the re-engineering process. We very rapidly ran into problems of scope and time. There is a tremendous amount of data written on the subject of re-engineering. The re-engineering data not only covered software, but also included business, hardware, and other re-engineering topics. The first problem we faced was separating out the relevant topics. After 'narrowing' the scope to software re-engineering, we were still faced with a daunting amount of information to review. We found a great deal of information presented on the general theory of re-engineering, but very little concrete information on the application of these theories. We were able to locate a methodology to try [30].

The first phase of this methodology is the extraction of data from the existing code. As noted by Byrne (30), tool support for this portion is a major necessity. With appropriate tools it is possible to statically analyze and document an extensive program in a relatively short period of time. He estimates a 150 to 1 time ratio difference between manual analysis to an automated tool approach. Based on our experience, we would concur with this evaluation. We were unable to locate any tool support to automate the extraction of basic calling trees or other data from the legacy system we were looking at. Although some were found for high order languages, there were very few listings for any Assembly language, and none of these were for the Motorola 6800. This we believe will be prevalent in most legacy systems. Most of the legacy systems will be in some form of Assembly language, but there are not enough in any one language to make it commercially feasible to develop a tool to automate this portion of the re-engineering process.

When faced with poorly designed software it is important to start the re-engineering work at the lowest level in which the design is masked. For example, the preliminary design may be adequate and it was in the execution of the design that the poor design decisions were made. However, it is more likely that the designers, knowing the constraints of the environment in which they have to operate, like available memory,

will design to make optimum use of the resources available. There are examples in which the software was written with self-modifying code in order to recover memory space used for initialization as scratch pads for data. However, when re-engineering the system these type of design decisions need to be abandoned and the software needs to be designed to the standards of the times.

With the TD-1271, a tremendous amount of time was used in just trying to understand and decipher the source code. The software, with little or no comments (see Appendix D) and poorly designed control structure, cannot be recommended for re-engineering. Time and effort could be expended to re-engineer the source code: It's not an impossible task, but it is not cost effective. Nor would the result have enhanced the supportability of the software.

The only reasonable approach for the TD1271B/U would be to start at the requirements documentation, establish a testing criteria for each requirement, and perform those tests on the existing system. These tests would demonstrate if there have been modifications to each requirement that would warrant further investigation to determine the current state of that requirement. Once the modifications to the requirements have been made and a suitable target system identified, the task of re-writing the software would be considered a small to medium effort when weighed against current Navy communications systems currently under development.

APPENDIX A

A. BIBLIOGRAPHY

1	"A Conceptual Foundation for Software Re-engineering", E. Byrne, IEEE 0-8186-2980-0/92
2	"A Framework for Reverse Engineering DoD Legacy Information Systems", P. Aiken, A. Muntz, R. Richards, IEEE 0-8186-3780-3/93
3	"A Software Re-engineering Process Model", E. Byrne, D. Gustafson, IEEE 0730-3315/92
4	"A Suzuki Class in Software Reengineering", Kozaczynski, Wojtek, IEEE-Software. v. 8 Jan. '91 p. 97-8
5	"An Economic Model to Estimate Software Rewriting and Replacement Times", Chan, Taizan; Chung, Siu-Leung; Ho, Teck-Hua, IEEE-Transactions-on-Software-Engineering. v. 22 Aug. '96 p. 580-98
6	"Analyzing Empirical Data from a Reverse Engineering Project", P. Fiore, F. Lanubile, and G. Visaggio, IEEE 0-8186-7111-4/95
7	"Automated Support of Software Maintenance", K. Bennett, Information And Software Technology, 33(1):74-85, Feb. 1991
8	"Cash Cow in the Tar Pit: Reengineering a Legacy System", W. Adolph, IEEE Software, May 1996
9	"Concept of the Re-engineering Life-Cycle", J. Manzella, B. Mutafelija, IEEE 0-8186-2697-6/92
10	"Design Recovery for Maintenance and Reuse", T. Biggerstaff, Computer, July '89
11	"DoD Legacy Systems; Reverse Engineering Data Requirements", Aiken, Peter; Muntz, Alice; Richards, Russ, Communications-of-the-ACM. v. 37 May '94 p. 26-41
12	"Improving Software Maintenance Through Measurement", Rombach, H.-Dieter; Ulery, Bradford-T, Proceedings-of-the-IEEE. v. 77 Apr. '89 p. 581-95
13	"Investigating Reverse Engineering Technologies for the CAS Program Understanding Project", Buss, E; De-Mori, R; Gentleman, W.-M, IBM-Systems-Journal. v. 33 no3 '94 p. 477-500
14	"Mandrake: A Tool for Reverse-Engineering IBM Assembly Code", P. Morris, R. Filman, IEEE Proceedings of WCRE '96
15	"Measurement: the key to application development quality", Walrad, C; Moss, E, IBM-Systems-Journal. v. 32 no3 '93 p. 445-60
16	"Planning the Reengineering of Legacy Systems", Sneed, Harry M., IEEE Software v 12 n 1 Jan 1995
17	"Program and Interface Slicing for Reverse Engineering", J. Beck D. Eichmann, IEEE 0270-5257/93
18	"Pros and Cons of Reverse Engineering", http://www.tailormade.com
19	"Recovering Design and Specifications from Source Code", Prywes, Noah, IEEE-Software. v. 13 Nov. '96 p. 109-11+
20	"Reengineering Procedural Into Data Flow Programs", P. Newcomb, IEEE 0-8186-7111-4/95
21	"Reengineering Definitions", http://www.stsc.hill.af.mil
22	"Retrieving Information from Data Flow Diagrams", G. Butler, P. Grogono, R. Shinghal, I. Tjandra, IEEE 0-8186-7111-4/95
23	"Reverse Engineering and Design Recovery: A Taxonomy", E. Chikofsky, J. Cross II,

	IEEE Software, Vol. 7, No. 1 Jan. 1990
24	"Reverse Engineering Programs via Dynamic Analysis", H. Ritsch, H. Sneed, Proceedings of WCRE '93
25	"Reverse Engineering through Formal Transformation: Knuths 'Polynomial Addition' Algorithm", Ward,-M.-P, The-Computer-Journal. v. 37 no9 '94 p. 795-813
26	"Reverse Engineering: Algorithms for Program Graph Production", Cimitile,-Aniello; De-Carlini,-Ugo, Software:-Practice-and-Experience. v. 21 May '91 p. 519-37
27	"Software Measurement: A Necessary Scientific Basis", Fenton,-Norman, IEEE-Transactions-on-Software-Engineering. v. 20 Mar. '94 p. 199-206
28	"Software Reengineering: A Quick History", R. Arnold, Communications of the ACM, May 1994
29	"Software Reengineering: Context and Definitions", R. Arnold, Software Reengineering, IEEE Computer Society Press, 1993. Pp. 25
30	"Software Reverse Engineering: A Case Study", Byrne,-Eric-J, Software:-Practice-and-Experience. v. 21 Dec. '91 p. 1349-64
31	"The Evolutionary Growth of Software Reengineering and the Decade Ahead", W. Ulrich, American Programmer, Oct 1990, pp 14-20
32	"Using an Enabling Technology to Reengineer Legacy Systems", Markosian,-Lawrence; Newcomb,-Philip; Brand,-Russell, Communications-of-the-ACM. v. 37 May '94 p. 58-70
33	"Using Data Abstraction to Guide the Restructuring of FORTRAN", R. Ellis, L. Liu, Class handout - source unknown
34	"Using Informal and Formal Techniques for the Reverse Engineeing of C Programs", G. Gannod, B. Cheng, IEEE Proceedings of WCRE '96
35	"Using metrics to evaluate software system maintainability", Coleman,-Don; Ash,-Dan; Lowther,-Bruce, Computer. v. 27 Aug. '94 p. 44-9
36	Untitled reverse engineering article, R. Waters, E. Chikofsky, Communications of the ACM, May 1994
37	"A Reverse Engineering Methodology to Reconstruct Hierarchical Data Flow Diagrams for Software Maintenance", P. Benedusi, A. Cimitile, U. De Carlini, Proceedings of the IEEE Conference on Softwar Maintenance, Oct 1998, pp 180-189
38	A Method for Re-Engineering in Structured Programs", A. Hevner, R. Linger, 22 nd Annual Hawaii International Conference on System Sciences, 1989, p. 1025-1034
39	"Object-Oriented Modeling and Design", J. Rumbaugh, M. Blaha, W. Premierlani, F. Eddy, W. Lorensen, Prentice Hall, 1991.
40	"Re-3' Re-Engineering , Restructuring, Reverse Engineering Part 1", E. Yourdon, American Programmer, Vol 2, No. 4, Apr 1989 p. 3-10, reprinted in R. Arnolds "Software Reengineering" IEEE Computer Society Press, 1992.
41	"A road map guide to Software Reengineering Technology", R. Arnold, Software Reengineering IEEE Computer Society Press, 1992.
42	"Software Engineering with Abstractions", Berzins and Luqi, Addison-Wesley Publishing Company, 1991
43	EE163-GA-OMI-010/E106-TD1271B, Operation And Maintenance Instructions, Organizational Level, Multiplexer TD-1271B/U, Change 2 dtd 07 June 1994,
44	FSCS-211 74, Command And Staff Manual For The UHF DAMA Subsystem, dtd 19 April 1985
45	FSCS-211-24-1 Revision A, UHF Demand Assigned Multiple Access (DAMA) Systems Interface Design Specification For The UHF Satcom Satellite Waveform, dtd 15 January 1985
46	FSCS-212-16 Revision D, UHF Demand Assigned Multiple Access (DAMA) System TD-1271B/U Multiplexer Program Performance Specification, dtd 01 September 1987

47	FSCS-212-17 Revision A, UHF Demand Assigned Multiple Access (DAMA) System TD-1271B/U Multiplexer Program Design Specification, dtd 01 September 1987
48	FSCS-212-26-2 (CH-1), Interface Control Document For The TD-1271B/U Multiplexer (Distributed Control Mode), dtd 30 September 1990
49	FSCS-212-7 Revision A, UHF Demand Assigned Multiple Access (DAMA) System TD-1271B/U Multiplexer Program Performance Specification Data Buffer Processor, dtd 15 May 83
50	FSCS-212-8 Revision A, UHF Demand Assigned Multiple Access (DAMA) System TD-1271B/U Multiplexer Program Design Specification Data Buffer Processor, dtd 01 September 1987
51	DOD-STD-1679A(Navy) Military Standard Software Development, dtd 22 October 1983

APPENDIX B

A. TD-1271B/U SUBROUTINE CALL TREE

Module Name	Located in	Is Called by	Located in
0,X		FPANEL	FP
		SRX	STPV06
		STX0	STPV06
ACK_CC_RCCOW	PRECLA	EXAM_CCOW_OVRHED	PRECLA
ADJUST_FCOUNT_GS	GS	DBP_MSG_INTRPRT	DCEXEC
		DBP_MSG_INTRPRT	PREALT
		EXAM_CCOW_OVRHED	PRECLA
ALARM_ON_GS	GS	FPANEL	FP
		BEEP_ALARM_GS	GS
ALPHA_ONOFF_A_GS	GS	FPANEL	FP
		GLIST_DISPLY_FP	FP
		RECALL_FP	FP
		DC_HOME_FREQ_FP	FP
		AC_HOME_FREQ_FP	FP
		QUEUE_TIME_FP	FP
		CMD_RESPONSE_FP	FP
		FREQ_TO_DSPLY_FP	FP
		KG_TIME_TO_DSPLY	FP
		STATUS_RPT_B_FP	FP
		ZEROIZE_FP	FP
		CONFER_CALL_FP	FP
		INFO_REQUEST_FP	FP
		PAGING_REQ_FP	FP
		KG_MEM_CHANGE_FP	FP
		CONFIG_CODE_FP	FP
		DISP_INFO_REQ_GS	GS
		LAMP_TEST_GS	GS
		LNK_TST_RES_GS	GS
		ERR_GS	GS
		SHOW_PORT_GS	GS
ALPHA_ONOFF_B_GS	GS	FPANEL	FP
		INFO_REPORT_FP	FP
		RECALL_FP	FP
		ZEROIZE_FP	FP
		PORT_RELEASE_FP	FP
		DC_HOME_FREQ_FP	FP
		AC_HOME_FREQ_FP	FP
		SELECT_PORT_FP	FP
		PREC_ENTER_FP	FP
		KG_MEM_FILL_FP	FP
		INFO_REQ_CHK_FP	FP
		EXIT_FP	FP
		LINK_TEST_REQ_FP	FP
		CALL_COMPLETE_FP	FP
		CHAN_CONTROL_FP	FP
		OUT_OF_SERV_FP	FP
		TO_PARITY_CALL_FP	FP
		FORMAT_FREQ_FP	FP
		CONFER_CALL_FP	FP
		KG_TIME_FP	FP

Module Name	Located in	Is Called by	Located in
ALPHA_ONOFF_B_GS	GS	CALL_CANCEL_FP	FP
		INFO_REQUEST_FP	FP
		PAGING_REQ_FP	FP
		KG_MEM_CHANGE_FP	FP
		STG_CONTROL_FP	FP
		CONFIG_CODE_FP	FP
ALPHA_ONOFF_B_GS	GS	LNK_TST_RES_GS	GS
		CALL_CC_GS	GS
		DISP_FCNT_GS	GS
AUDIO_ALARM_01	FP	FPANEL	FP
AUDIO_ALARM_02	FP	INFO_REQ_CHK_FP	FP
AUDIO_ALARM_OFF	FP	FPANEL	FP
		KG_MEM_FILL_FP	FP
BCD_TO_BIN_GS	GS	INFO_REQUEST_FP	FP
		YZ_TO_BIN_FP	FP
		AC_HOME_FREQ_FP	FP
		VWXYZ_TO_BIN_FP	FP
BCD_TO_YZ_FP	FP	FPANEL	FP
		FREQ_TO_DSPLY_FP	FP
		DC_HOME_FREQ_FP	FP
BEEP_ALARM_GS	GS	FPANEL	FP
		INFO_REPORT_FP	FP
		PORT_RELEASE_FP	FP
		DC_HOME_FREQ_FP	FP
		AC_HOME_FREQ_FP	FP
		SELECT_PORT_FP	FP
		PREC_ENTER_FP	FP
		KG_MEM_FILL_FP	FP
		CMD_RESPONSE_FP	FP
		PRE_DCKA_CHK_FP	FP
		CHAN_CONTROL_FP	FP
		LINK_TEST_REQ_FP	FP
		ZEROIZE_FP	FP
		STATUS_RPT_B_FP	FP
		CALL_COMPLETE_FP	FP
		OUT_OF_SERV_FP	FP
		TO_PARITY_CALL_FP	FP
		FORMAT_PARITY_FP	FP
		CONFER_CALL_FP	FP
		KG_TIME_FP	FP
		CALL_CANCEL_FP	FP
		INFO_REQUEST_FP	FP
		PAGING_REQ_FP	FP
		EXTEND_XMIT_FP	FP
		KG_MEM_CHANGE_FP	FP
		STG_CONTROL_FP	FP
		CONFIG_CODE_FP	FP
		PORT_DETECT_GS	GS
		LNK_TST_RES_GS	GS
		ERR_GS	GS
BIN_TO_BCD_CL_GS	GS	FPANEL	FP
		DC_HOME_FREQ_FP	FP
		AC_HOME_FREQ_FP	FP
		QUEUE_TIME_FP	FP
		DISCOTIME_DISPLAY	FP
		FREQ_TO_DSPLY_FP	FP
		KG_TIME_TO_DSPLY	FP

Module Name	Located in	Is Called by	Located in
BIN_TO_BCD_GS	GS	FPANEL	FP
		RECALL_FP	FP
		INFO_REQ_CHK_FP	FP
		FREQ_CODE_CHK_FP	FP
		GLIST_DISPLY_FP	FP
		LNK_TST_RES_GS	GS
BIT_RATE_FIND	DCMOW	CALL_REQUEST	DCMOW
BITE_CLEAR	DCMOW	RRU_INTRPRT	DCMOW
		CALL_REQUEST	DCMOW
		DATA_XFER	DCMOW
		GLIST_DISP	DCMOW
		CALL_COMPLETE	DCMOW
		INFO_REPORT	DCMOW
		RRU_INTRPRT	PREINT
		RRU_STATUS_RPTB	PREINT
		DATA_XFER	PREINT
		GLIST_DISPLAY_05	PREINT
		CALL_COMPLETE	PREINT
		RRU_PRTY_OUT_05	PREINT
		INFO_REPORT_05	PREINT
		CALL_REQ_05	PREINT
		CONFIG_ENTRY_05	PREINT
		RRU_CALLCAN_05	PREINT
		GLIST_ENTDEL_10	PREINT
		PAGING_RCOW	PREINT
		CONF_REQUEST	PREINT
BITRAT	DBPV4	RBIJ	DBPV4
		RBIT	DBPV4
BLINK_SCULITE_GS	GS	EXEC_DCEXEC	DCEXEC
		EXEC_PREALT	PREALT
		CHECKSUM_CHK_GS	GS
BRX	STPV06	CONFIG	STPV06
BSB	STPV06	EXEC	STPV06
		PSA	STPV06
BTX	STPV06	CONFIG	STPV06
CALC_PARITY_GS	GS	XFER_CCOW_TO_DBP	DCEXEC
		IN_CCOW_FROM_DBP	DCEXEC
		ENCRYPT_CCOW	DCEXEC
		DECRYPT_CCOW	DCEXEC
		XFER_RCOW_DBP_GS	GS
		DECRYPT_RCOW_GS	GS
		READ_DBP_RCOW_GS	GS
		ENCRYPT_RCOW_GS	GS
		IN_CCOW_FROM_DBP	PREALT
		ENCRYPT_CCOW	PREALT
		DECRYPT_CCOW	PREALT
		XFER_OW	PREALT
CALL_CC_GS	GS	PORT_DETECT_GS	GS
CALL_COMPLETE_FP	FP	JUMP_COMPLETE_GS	GS
CALL_QUE_UPDATE	PREINT	CALL_IN_QUE_ACK	PREINT
		CALL_IN_QUE	PREINT
CALL_REQ_DCMOW	FP	CALL_REQUEST	DCMOW
		TO_PARITY_CALL_FP	FP
CC_EXEC_DCCNTR	DCCNTR	EXEC_DCEXEC	DCEXEC
CC_EXEC_PRECLA	PRECLA	EXEC_PREALT	PREALT

Module Name	Located in	Is Called by	Located in
CCOW_INTP_DCMOW	DCMOW	EXEC_DCEXEC	DCEXEC
		USEREXEC	DCEXEC
CCOW_INTP_PREINT	PREINT	USEREXEC	PREALT
CHECK_A_RANGE	DCMOW/PREINT	PORT_DISCONNECT	DCMOW
CHECK_B_RANGE	PREINT	PORT_DISCONNECT	PREINT
		PORT_DISCONNECT	DCMOW
CHECK_C_RANGE	DCMOW/PREINT	PORT_DISCONNECT	PREINT
		PORT_DISCONNECT	DCMOW
CHECKSUM_CHK_GS	GS	PORT_DISCONNECT	PREINT
		INIT_DCEXEC	DCEXEC
CHK_FRAME_COUNT	PRECLA	INIT	PREALT
		FRAME_CHK_PRECLA	PRECLA
CHK_GLIST	PREINT	SLOT_DISCONNECT	PREINT
CLEAR	DBPV4	CALL_WAITING	PREINT
		HOME_CHAN_CHANG	PREINT
		SLOT_CONNECT	PREINT
		FORMAT_CHANGE	PREINT
		PWRON	DBPV4
CLEAR_CONT_COUNT	PREINT	SETUP	DBPV4
		BUSY_ACK	PREINT
		CALL_IN_QUEUE_ACK	PREINT
		USER_OUT_ACK	PREINT
CLEAR_DISP_GS	GS	CALL_ACK	PREINT
		FPANEL	FP
CLEAR_GNUM	PREINT	SHOW_PORT_GS	GS
		RPT_GLIST1	PREINT
		RPT_GLIST2	PREINT
		RPT_GLIST3	PREINT
		RTP_GLIST4	PREINT
CLEAR_GUARD_LIST	DCMOW	FRAME_FMT_UPDATE	DCMOW
CLEAR_LITES_TIME	PREINT	CALL_CANCEL	PREINT
CLEAR_QUEUE_TIME	PREINT	SLOT_CON_UPDATE	PREINT
CLEAR_SLOTS_GS	GS	RRU_CALLCAN_15	PREINT
		PORT_DISCONNECT	DCMOW
		CHAN_CONTROL_FP	FP
		STATUS_PRECLA	PRECLA
		ZEROIZE	PREINT
CLR_RRU_05	PREALT	PORT_DISCONNECT	PREINT
		RRU_MSG_UPDATE	DCEXEC
CLR_RRU_PREC_DSP	DCEXEC/PREALT	RRU_MSG_UPDATE	DCEXEC
		RRU_MSG_UPDATE	PREALT
CLRTIM	PREINT	BUSY_ACK	PREINT
		USER_OUT_ACK	PREINT
		SLOT_CON_UPDATE	PREINT
		CALL_CANCEL	PREINT
		PARTY_OUT_OF_SYNC	PREINT

Module Name	Located in	Is Called by	Located in
CMD_RESPONSE_FP	FP	ZEROIZE_FP	FP
		INFO_REPORT_FP	FP
		TO_PARITY_CALL_FP	FP
		FORMAT_FREQ_FP	FP
		CONFER_CALL_FP	FP
		DC_HOME_FREQ_FP	FP
		AC_HOME_FREQ_FP	FP
		KG_TIME_FP	FP
		INFO_REQUEST_FP	FP
		GLIST_ENTER_FP	FP
		PAGING_REQ_FP	FP
		KG_MEM_CHANGE_FP	FP
		GLIST_DELETE_FP	FP
		STG_CONTROL_FP	FP
		CONFIG_CODE_FP	FP
		CHAN_CONTROL_FP	FP
COMMAND_FRAME	PREINT	CCOW_INTP_DCMOW	DCMOW
		CCOW_INTP_05	PREINT
COMP_FRAME_GS	GS	FRAME_CHK_DCCNTR	DCCNTR
		FRAME_CHK_PRECLA	PRECLA
		MASTER FRAME	PREINT
CONFIG_CODE_GS	GS	INFO_REQ_CHK_FP	FP
CONS	STPV06	CONFIG	STPV06
CONT_CHK_PREINT	PREINT	RCOW_PREP_XMIT	PREALT
		CLEAR_CONT_COUNT	PREINT
		READ_CALL_ACK	PREINT
CORRECT_ONE_BCD	GS	BCD_TO_BIN_GS	GS
CORRECT_TWO_BCD	GS	BIN_TO_BCD_GS	GS
CYCLELITES_DCMOW	DCMOW	ZEROIZE	DCMOW
		TS_PREP	DCMOW
		INFO_REQUEST_GS	GS
DBP_MSG_INT_30	DCEXEC/PREALT	EXEC PREALT	PREALT
DBP_MSG_INT_35	DCEXEC/PREALT	EXEC DCEXEC	DCEXEC
DBP_MSG_INTRPRT	DCEXEC/PREALT	USEREXEC	DCEXEC
		USERPREAC	DCEXEC
		USERPREAC	PREALT
		USEREXEC	PREALT
DCKPORTOFF_GS	GS	INFO_REQUEST_GS	GS
		PORT_DETECT_GS	GS
DECODE_FP	FP	FPANEL	FP
		KG_MEM_FILL_FP	FP
DECRYPT_CCOW	DCEXEC/PREALT	DBP_MSG_INTRPRT	DCEXEC
		DBP_MSG_INTRPRT	PREALT
DECRYPT_RCOV_GS	GS	DBP_MSG_INTRPRT	DCEXEC
		DBP_MSG_INTRPRT	PREALT
DEL_FROM_GLIST	PREINT	DELET_GUARD_LIST	PREINT
		GLIST_ENTDEL_30	PREINT
DELAY_TOGGLE_GS	GS	STATUS_DCCNTR	DCCNTR
		CC_EXEC_PRECLA	PRECLA
		RCV_CCOW_BLOCK	PRECLA
		STATUS_PRECLA	PRECLA
DID	STPV06	RNGEXC	STPV06
		LTEXEC	STPV06

Module Name	Located in	Is Called by	Located in
DISCO_SLOT_CHK	DCMOW/PREINT	CHECK_A_RANGE	DCMOW
		CHECK_B_RANGE	DCMOW
		CHECK_C_RANGE	DCMOW
		CHECK_A_RANGE	PREINT
		CHECK_B_RANGE	PREINT
		CHECK_C_RANGE	PREINT
DISCOTIME_DISPLY	FP	FPANEL	FP
DISP_FCNT_GS	GS	RECALL_FP	FP
		ZEROIZE_FP	FP
		CHAN_CONTROL_FP	FP
		DC_HOME_FREQ_FP	FP
		AC_HOME_FREQ_FP	FP
		SELECT_PORT_FP	FP
		CALL_COMPLETE_FP	FP
		STATUS_RPT_B_FP	FP
		GLIST_DISPLY_FP	FP
		LINK_TEST_REQ_FP	FP
		INFO_REPORT_FP	FP
		TO_PARITY_CALL_FP	FP
		FORMAT_FREQ_FP	FP
		CONFER_CALL_FP	FP
		KG_TIME_FP	FP
		CALL_CANCEL_FP	FP
		INFO_REQUEST_FP	FP
		GLIST_ENTER_FP	FP
		PAGING_REQ_FP	FP
		KG_MEM_CHANGE_FP	FP
		GLIST_DELETE_FP	FP
		STG_CONTROL_FP	FP
		CONFIG_CODE_FP	FP
		OUT_OF_SERV_FP	FP
		LNK_TST_RES_GS	GS
DISP_INFO_REQ_GS	GS	FPANEL	FP
DISP_VWXYZ_GS	GS	FPANEL	FP
DISPLAY_EEEEE_GS	GS	RECALL_FP	FP
		LNK_TST_RES_GS	GS
		GLIST_DISPLY_FP	FP
DISPLY_TO_BCD_GS	GS	ERR_GS	GS
		FORMAT_FREQ_FP	FP
		AC_HOME_FREQ_FP	FP
		VWXYZ_TO_BIN_FP	FP
		YZ_TO_BIN_FP	FP
		WX_TO_BCD_FP	FP
		YZ_TO_BCD_FP	FP
		CONFIG_CODE_FP	FP
ENCRYPT_CCOW	DCEXEC/PREALT	EXEC_DCEXEC	DCEXEC
ENCRYPT_RCOW_GS	PREALT	EXEC_PREALT	PREALT
		RCOW_PREP_XMIT	DCEXEC
		RCOW_PREP_XMIT	PREALT
EQUIPLED_REDO	FP	FPANEL	FP

Module Name	Located in	Is Called by	Located in
ERR_GS	GS	ZEROIZE_FP	FP
		INFO_REPORT_FP	FP
		TO_PARITY_CALL_FP	FP
		FORMAT_FREQ_FP	FP
		CONFER_CALL_FP	FP
		KG_TIME_FP	FP
		INFO_REQUEST_FP	FP
		GLIST_ENTER_FP	FP
		PAGING_REQ_FP	FP
		EXTEND_XMIT_FP	FP
		GLIST_DELETE_FP	FP
		GLIST_DISPLAY_FP	FP
EVEN_PORT_STORE	PREINT	RRU_STATUS_RPTB	PREINT
EXAM_CCOW_OVRHED	PRECLA	RCV_CCOW_BLOCK	PRECLA
EXIT_FP	FP	TIME_CHK_DCCNTR	DCCNTR
FIND_GLIST	PREINT	ENTER_GUARD_LIST	PREINT
		DELET_GUARD_LIST	PREINT
FIND_GLIST_05	PREINT	GLIST_DISPLAY_05	PREINT
		GLIST_ENTDEL_10	PREINT
		GLIST_ENTDEL_30	PREINT
FIND_PREMPT_ADR	PREINT	PREMPTCOUNT REDO	PREINT
		SLOT_CON_UPDATE	PREINT
		RRU_DISCO	PREINT
FIND_RRU_BUF_GS	GS	SETUPI	DCMOW
		CALL_REQUEST	DCMOW
		LITES_OFF DCMOW	DCMOW
		DATA_XFER	DCMOW
		LNK_TST_LAMP_GS	GS
		LNK_TST_RES_GS	GS
		RRU_ACPT_ON_GS	GS
		INFO_REQUEST_GS	GS
		RRU_DISCO	PREINT
		READ_CALL_ACK	PREINT
		CALL_IN_QUE_ACK	PREINT
		USER_OUT_ACK	PREINT
		RRU_OUT_ADR	PREINT
		SLOT_CON_UPDATE	PREINT
		CALL_CANCEL	PREINT
		CALL_WAITING	PREINT
		CALL_IN_QUE	PREINT
		CCOW_DATA_XFER	PREINT
		PARTY_OUT_OF_SVC	PREINT
		LITES_OFF PREINT	PREINT
		RRU_CALL_CAN_05	PREINT
		DATA_XFER	PREINT
FIND_RRU_QUE	PREINT	DEDICATED_SLOT	PREINT
		BUSY_ACK	PREINT
		CALL_WAITING	PREINT
		FIND_RCOW_QUE	PREINT
FIND_RRUQUE_ADR	PREINT	CALL_QUE_UPDATE	PREINT
		OUT_SVC_UPDATE	PREINT
		CLEAR_QUE_TIME	PREINT

Module Name	Located in	Is Called by	Located in
FIXA_GS	GS	FPANEL	FP
		CONFER_CALL_FP	FP
		QUEUE_TIME_FP	FP
		DISCOTIME_DISPLAY	FP
		KG MEM_FILL_FP	FP
		FREQ_TO_DSPLY_FP	FP
		PAGING_REQ_FP	FP
		KG MEM_CHANGE_FP	FP
		LINK_TEST_REQ_FP	FP
FIXAB_GS	GS	FIXAB_GS	GS
		FPANEL	FP
		SHOW_YZ_GS	GS
		SHOW_WX_GS	GS
FIXB_GS	GS	FIXA_GS	GS
		QUEUE_TIME_FP	FP
		DISCOTIME_DISPLAY	FP
		CMD_RESPONSE_FP	FP
		FREQ_TO_DSPLY_FP	FP
FMT_GLIST_RCW_05	PREINT	DISPLY_TO_BCD_GS	GS
		SET_GLIST_PTR	PREINT
		SET_GLIST_PTR	PREINT
		SET_GLIST_PTR	PREINT
		CLEAR_GNUM	PREINT
		FORMAT_CHANGE	PREINT
		RPT_GLISTI	PREINT
		FP_RCOW	DCEXEC/PREALT
		USEREXEC	DCEXEC
FPANEL_FP	FP	USERPREAC	DCEXEC
		USERPREAC	PREALT
		USEREXEC	PREALT
		USEREXEC	DCEXEC
		USERPREAC	DCEXEC
FRAME_CHK_DCCNTR	DCCNTR	USEREXEC	PREALT
		USEREXEC	DCEXEC
		USERPREAC	DCEXEC
		DBP_MSG_INTRPRT	DCEXEC
FRAME_CHK_PRECLA	PRECLA	USEREXEC	PREALT
		USERPREAC	PREALT
		DBP_MSG_INTRPRT	PREALT
FRAME_FMT_UPDATE	DCMOW	MASTER_FRAME	DCMOW
FREEZE_DISPLAY	FP	FORMAT_FREQ_FP	FP
		DC_HOME_FREQ_FP	FP
		KG_TIME_FP	FP
		KG_MEM_CHANGE_FP	FP
FREQ_CODE_CHK_5	FP	KG_TIME_FP	FP
FREQ_CODE_CHK_FP	FP	FORMAT_FREQ_FP	FP
		DC_HOME_FREQ_FP	FP
		FORMAT_FREQ_FP	FP
FREQ_TO_DSPLY_FP	FP	FORMAT_FREQ_FP	FP
FSLOT	STPV06	SRX	STPV06
GET_RANDOM_NR_GS	GS	RCOW_PREP_XMIT	PREALT
		COMP_RRU_PREC	PREINT

Module Name	Located in	Is Called by	Located in
GET_RCOW_ADR_GS	GS	DATA_XFER	DCMOW
		INFO_REQUEST	DCMOW
		ANY_MSG_IN_QUE	DCMOW
		FRAME_CHK_PRECLA	PRECLA
		RRU_STATUS_RPTB	PREINT
		DATA_XFER	PREINT
		CALL_COMPLETE	PREINT
		RRU_PRTY_OUT_05	PREINT
		INFO_REPORT_05	PREINT
		RRU_CALLCAN_05	PREINT
		RRU_CALL_CAN_15	PREINT
		PAGING_05	PREINT
		FIND_RRU_QUE	PREINT
		ANY_MSG_IN_QUE	PREINT
		CONFIR_LIST RTP	PREINT
		DEDICATED_SLOT	PREINT
		CLRTIM	PREINT
		CALL_REQ_05	PREINT
		CONF_REQ_05	PREINT
GET_RRU_INADR_GS	GS	RRU_INTRPRT	DCMOW
		CALL REQUEST	DCMOW
		DATA-XFER	DCMOW
		GLIST_DISP	DCMOW
		CALL_COMPLETE	DCMOW
		INFO_REQUEST	DCMOW
		RRU_INTRPRT	PREINT
		CONF REQUEST	PREINT
		CALL REQUEST	PREINT
		RRU_STATUS_RPTB	PREINT
		DATA_XFER	PREINT
		GLIST_DISPLAY	PREINT
		CALL_COMPLETE	PREINT
		RRU_PRTY_OUT_05	PREINT
		INFO_REPORT	PREINT
		RRU_CALL_CANCEL	PREINT
		GLIST_ENTR_DEL	PREINT
		GLIST_ENTDEL_05	PREINT
		PAGING_RCOW	PREINT
		CONFIG ENTRY	PREINT
GET_TIME_CODE	FP	OUT_OF_SERV_FP	FP
		CONFER_CALL_FP	FP
GLIST_COMPARE	PREINT	GLIST_DISP_05	PREINT
		CHK_GLIST	PREINT
GLIST_COUNTER	PREINT	ENTER_GUARD_LIST	PREINT
		GLIST_ENTDEL_10	PREINT
		GLIST_ENTDEL_30	PREINT
		DELET_GUARD_LIST	PREINT
HI_GUARD_CHK_FP	FP	ZEROIZE_FP	FP
		INFO_REPORT_FP	FP
		TO_PARITY_CALL_FP	FP
		CONFER_CALL_FP	FP
		INFO_REQUEST_FP	FP
		GLIST_ENTER_FP	FP
		PAGING_REQ_FP	FP
IN_CCOW_FROM_DBP	DCEXEC	DBP_MSG_INTRPRT	DCEXEC
		DBP_MSG_INTRPRT	PREALT

Module Name	Located in	Is Called by	Located in
INFO_REPORT_FP	FP	CALL_CC_GS	GS
INFO_REQ_05_GS	GS	HOME_CHAN_05	PREINT
INFO_REQ_CHK_45	FP	SELECT_PORT_FP	FP
INFO_REQ_CHK_50	FP	PORT_RELEASE_FP	FP
INFO_REQ_CHK_FP	FP	FPANEL	FP
INIT	PREALT/DBPV4 (LABEL)	PWRON	DBPV4
INITIAL_ENTRY	PREINT	TDMA	DBPV4
		RRU_STATUS_RPTB	PREINT
		DATA_XFER	PREINT
		CALL_COMPLETE	PREINT
		RRU_PRTY_OUT_05	PREINT
		INFO_RPT_05	PREINT
		RRU_CALL_CAN_15	PREINT
		PAGING_05	PREINT
		STATUS_RPT_A	PREINT
		LINK_RESULT	PREINT
		FORMAT_GLIST_RCW	PREINT
		CALL_REQ_05	PREINT
		CONF_REQ_05	PREINT
IO1LF	DBPV4	IO1A	DBPV4
IO1LS	DBPV4	IO1A	DBPV4
IO2LF	DBPV4	IO2A	DBPV4
IO2LS	DBPV4	IO2A	DBPV4
IO3LF	DBPV4	IO3A	DBPV4
IO3LS	DBPV4	IO3A	DBPV4
IO4LF	DBPV4	IO4A	DBPV4
IO4LS	DBPV4	IO4A	DBPV4
JUMP_COMPLETE_GS	GS	DCKPORTOFF_GS	GS
JUMP_SELECT_1_GS	GS	DCKPORTOFF_GS	GS
JUMP_SELECT_2_GS	GS	SHOW_PORT_GS	GS
		CALL_CC_GS	GS
		DCKPORTOFF_GS	GS
KB_TIME_INIT_GS	GS	SHOW_PORT_GS	GS
		CALL_CC_GS	GS
		ZEROIZE_FP	FP
		OUT_OF_SERV_FP	FP
		STATUS_RPT_B_FP	FP
		GLIST_DISPLY_FP	FP
		LINK_TEST_REQ_FP	FP
		INFO_REQUEST_FP	FP
		TO_PARITY_CALL_FP	FP
		CONFER_CALL_FP	FP
		CALL_CANCEL_FP	FP
		INFO_REQUEST_FP	FP
		GLIST_ENTER_FP	FP
		PAGING_REQ_FP	FP
		CONFIG_CODE_FP	FP
		CALL_COMPLETE_FP	FP
		EXEC_DCEXEC	DCEXEC
		EXEC_PREALT	PREALT
		ENCRYPT_CCOW	PREALT

Module Name	Located in	Is Called by	Located in
KG_INITIAL_GS_05	GS	ENCRYPT_CCOW	DCEXEC
		DECRYPT_CCOW	DCEXEC
		DECRYPT_RCOW_GS	GS
		ENCRYPT_RCOW_GS	GS
		ENCRYPT_CCOW	PREALT
		DECRYPT_CCOW	PREALT
KG_LOAD_GS	GS	USERPREAC	DCEXEC
		ENCRYPT_CCOW	DCEXEC
		USERPREAC	PREALT
KG_MEM_CHANGE_GS	GS	EXEC_DCEXEC	DCEXEC
		TS_PREP_DCEXEC	DCEXEC
		EXEC_PREALT	PREALT
		TS_PREP_PREALT	PREALT
		KG_LOAD_GS	GS
KG_MEM_FILL_20	FP	KG_MEM_FILL_FP	FP
KG_MEM_FILL_FP	FP	FPANEL	FP
KG_MEM_TEST_GS	GS	EXAM_CCOW_OVRHED	PRECLA
		KG_MEM_CHANGE_GS	GS
KG_READY_GS	GS	EXAM_CCOW_OVRHED	PRECLA
		KG_RESET_GS	GS
		KG_INITIAL_GS	GS
KG_RESET_GS	GS	EXEC_DCEXEC	DCEXEC
		TS_PREP_DCEXEC	DCEXEC
		ENCRYPT_CCOW	DCEXEC
		DECRYPT_CCOW	DCEXEC
		DECRYPT_RCOW_GS	GS
		ENCRYPT_RCOW_GS	GS
		KG_MEM_CHANGE_GS	GS
		EXEC_PREALT	PREALT
		TS_PREP_PREALT	PREALT
		ENCRYPT_CCOW	PREALT
		DECRYPT_CCOW	PREALT
		KG_TIME_FP	FP
		KG_TIME_FP	FP
		KG_ZEROIZE_GS	GS
KG_TIME_75	FP	KG_TIME_FP	FP
		KG_TIME_FP	FP
KG_ZEROIZE_GS	GS	STATUS_DCCNTR	DCCNTR
		STATUS_PRECLA	PRECLA
LD1	DBPV4	IO1A	DBPV4
LD2	DBPV4	IO2A	DBPV4
LD3	DBPV4	IO3A	DBPV4
LD4	DBPV4	IO4A	DBPV4
LITES_OFF_DCMOW	DCMOW	STATUS_DCCNTR	DCCNTR
		ZEROIZE	DCMOW
		FRAME_FMT_UPDATE	DCMOW
		CHAN_CONTROL_FP	FP
LITES_OFF_GS	GS	RECALL_FP	FP
		INFO_REQUEST_FP	FP
		PORT_RELEASE_FP	FP
		INFO_REQ_CHK_FP	FP
		CALL_COMPLETE_FP	FP
		LINK_TEST_REQ_FP	FP
		ZEROIZE_FP	FP
		STATUS_RPT_B_FP	FP
		GLIST_DISPLY_FP	FP
		CHAN_CONTROL_FP	FP
		OUT_OF_SERV_FP	FP

Module Name	Located in	Is Called by	Located in
LITES_OFF_PREINT	PREINT	STATUS_PRECLA	PRECLA
		FORMAT_UPDATE	PREINT
		NUFMT_TO_STP	PREINT
LNK_TST_LAMP_GS	GS	TO_PARITY_CALL_FP	FP
		FORMAT_FREQ_FP	FP
		PORT_RELEASE_FP	FP
		DC_HOME_FREQ_FP	FP
		AC_HOME_FREQ_FP	FP
		QUEUE_TIME_FP	FP
		SELECT_PORT_FP	FP
		CONFER_CALL_FP	FP
		KG_TIME_FP	FP
		CALL_CANCEL_FP	FP
		INFO_REQUEST_FP	FP
		GLIST_ENTER_FP	FP
		PAGING_REQ_FP	FP
		EXTEND_XMIT_FP	FP
		KG_MEM_CHANGE_FP	FP
		GLIST_DELETE_FP	FP
		STG_CONTROL_FP	FP
		CONFIG_CODE_FP	FP
		LAMP_TEST_GS	GS
LO_GUARD_CHK_FP	FP	GLIST_ENTER_FP	FP
LOAD_KG_VARS_GS	GS	ENCRYPT_CCOW	DCEXEC
		DECRYPT_CCOW	DCEXEC
		ENCRYPT_RCOW_GS	GS
		ENCRYPT_RCOW_GS	GS
		ENCRYPT_CCOW	PREALT
		DECRYPT_CCOW	PREALT
LTRXEX	STPV06	EXEC	STPV06
MASTER_FRA_05	PREINT	MASTER_FRAME	PREINT
MASTER_FRA_10	PREINT	MASTER_FRAME	PREINT
MASTER_FRAME	DCMOW/PREINT/DCCNTR	CCOW_INTP_DCMOW	DCMOW
		CCOW_INTP_PREINT	PREINT
MOVE_FRAME_GS	GS	MASTER_FRAME	DCCNTR
		FRAME_CHK_DCCNTR	DCCNTR
MOVE_GS	GS	SPECIAL_CCOW	DCCNTR
		RCOW_INTP_DCCNTR	DCCNTR
		INIT_DCEXEC	DCEXEC
		MOVE_TMP_TO_VARS	DCEXEC
		MOVE_VARS_TO_TMP	DCEXEC
		STP_MSG_INTRPRT	DCEXEC
		INIT	PREALT
		MOVE_VARS_TO_TMP	PREALT
		MOVE_TMP_TO_VARS	PREALT
		STP_MSG_INTRPRT	PREALT
		RCOW_INTP_PRECLA	PRECLA
MOVE_TMP_TO_VARS	DCEXEC/PREALT	USERPREAC	PREALT
		USERPREAC	DCEXEC
MOVE_VARS_TO_TMP	DCEXEC/PREALT	USERPREAC	DCEXEC
		USERPREAC	PREALT
NET_NUM_XCHNG	DCEXEC	DECRYPT_CCOW	DCEXEC
NUFMT_TO_STP	PREINT	MASTER_FRAME	PREINT
		FORMAT_CHANGE	PREINT
ODD_PORT_STORE	PREINT	RRU_STATUS_RPTB	PREINT

Module Name	Located in	Is Called by	Located in
OFFSET_BY_3_GS	GS	CC_EXEC_DCCNTR	DCCNTR
		COMMAND_FRAME	DCMOW
OFFSET_BY_6_GS	GS	CC_EXEC_DCCNTR	DCCNTR
		COMMAND_FRAME	DCMOW
OUT_SVC_UPDATE	PREINT	USER_OUT_ACK	PREINT
		PART_OUT_OF_SVC	PREINT
PLACE_CALL_PRTY	PREINT	CONF_REQ_05	PREINT
PLACE_PORT	PREINT	CONFER_LIST_RPT	PREINT
		STATUS_RPT_B	PREINT
POINT_TO_GLIST1	PREINT	SET_GLIST_PTR	PREINT
POINT_TO_GLIST2	PREINT	SET_GLIST_PTR	PREINT
POINT_TO_GLIST3	PREINT	SET_GLIST_PTR	PREINT
POINT_TO_GNUM	PREINT	POINT_TO_GLIST1	PREINT
		POINT_TO_GLIST2	PREINT
		POINT_TO_GLIST3	PREINT
PORT_DETECT_GS	GS	DBP_MSG_INTRPRT	DCEXEC
PORT_DISCO_UPDATE	PREINT	RRU_UPDATE	PREINT
		ZEROIZE	PREINT
PORT_DISCONNECT	DCMOW	FRAME_FMT_UPDATE	DCMOW
		FORMAT_UPDATE	PREINT
		NUFMT_TO_STP	PREINT
PORT_RELEASE_15	FP	FPANEL	FP
PORT_RELEASE_FP	FP	DCKPORTOFF_GS	GS
		SHOW_PORT_GS	GS
		CALL_CC_GS	GS
PORTCCADR_05	PREINT	CONFIG_ENTRY_05	PREINT
PORTCCADR_PREINT	PREINT	CONFIG_CODE_FP	FP
		SLOT_CON_UPDATE	PREINT
		RRU_STATUS_RPTB	PREINT
		CALL_REQ_05	PREINT
		CONF_REQ_05	PREINT
		CONFIG_ENTRY_05	PREINT
PRE_DCKA_CHK_FP	FP	TO_PARTY_CALL_FP	FP
PREC_CONVERT_FP	FP	FPANEL	FP
		TO_PARITY_CALL_FP	FP
PREC_ENTER_25	FP	FORMAT_FREQ_FP	FP
PREC_ENTER_30	FP	GLIST_DISPLY_FP	FP
PREC_ENTER_FP	FP	OUT_OF_SERV_FP	FP
		TO_PARITY_CALL_FP	FP
		CONFER_CALL_FP	FP
PREC_TO_RCOW_GS	GS	DATA_XFER	DCMOW
		DATA_XFER	PREINT
		RRU_PRTY_OUT	PREINT
		CALL_REQ_05	PREINT
		CONF_REQ_05	PREINT
PREEMPT_LITE	PRINT	RRU_DISCO	PREINT
PREEMPT_UPDATE	PRECLA	TIME_CHK_PRECLA	PRECLA
PREMPTCOUNT_05	PREINT	PREMPTCOUNT_REDO	PREINT
PREMPTCOUNT_REDO	PREINT	SLOT_CON_UPDATE	PREINT
		SLOT_DISCONNECT	PREINT
		CALL_COMPLETE	PREINT
PRSN	STPV06	SRX	STPV06
PRT_CNFIG_CHNG	PREINT	RRU_STATUS_RPTB	PREINT
PSA	STPV06	EXEC	STPV06
		CONFIG	STPV06
PTSN	STPV06	STX	STPV06

Module Name	Located in	Is Called by	Located in
PVR	STPV06	SRX	STPV06
		BRX	STPV06
PVRA	STPV06	SRX	STPV06
PVT	STPV06	STX	STPV06
		BTX	STPV06
RAMT	DBPV4/STPV06	COLDS	STPV06
RCCOW_PORTADR_GS	GS	DATA_XFER	DCMOW
		INFO_REPORT	DCMOW
		RRRU_STATUS_RPTB	PREINT
		DATA_XFER	PREINT
		CALL_COMPLETE	PREINT
		RRU_PRTY_OUT_05	PREINT
		INFO_RPT_05	PREINT
		RRU_CALLCAN_15	PREINT
		PAGING_05	PREINT
		STATUS_RPT_A	PREINT
		LINK_RESULT	PREINT
		CALL_REQ_05	PREINT
		CONF_REQ_05	PREINT
RCOW_ASSIGN_FMT	PREINT	CALL_ACK_05	PREINT
		RCOW_ASSIGN_05	PREINT
RCOW_INTP_DCCNTR	DCCNTR	USEREXEC	DCEXEC
RCOW_INTP_PRECLA	PRECLA	USEREXEC	PREALT
RCOW_OUT_PREINT	PRINT	USEREXEC	PREALT
RCOW_PREP_XMIT	DCEXEC/PREALT	USEREXEC	DCEXEC
		USEREXEC	PREALT
RCOW_XMIT_DCMOW	DCMOW	USEREXEC	DCEXEC
RCV_CCOW_BLOCK	PRECLA	CC_EXEC_PRECLA	PRECLA
READ_ACIA	GS	TEST_ACIA	GS
READ_CALL_ACK	PREINT	MASTER_FRAME	PREINT
		COMMAND_FRAME	PREINT
READ_DBP_RCOW_GS	GS	DBP_MSG_INTRPRT	DCEXEC
		DBP_MSG_INTRPRT	PREALT
REDO_DBP_PORT_GS	GS	DISCO_SLOT_CHK	DCMOW
		CALL_REQUEST	DCMOW
		CALL_COMPLETE	DCMOW
		INIT	PREALT
		RRU_TURNOFF	PRECLA
		SLOT_CON_UPDATE	PREINT
		PORT_DISCONNECT	PREINT
		CLEAR_SLOTS_GS	GS
REDO_RRU1_GS	GS	REDO_RRUS_DCMOW	DCMOW
		REDO_RRUS_PREINT	PREINT
		WRITE_ACIA	GS
REDO_RRU2_GS	GS	REDO_RRUS_DCMOW	DCMOW
		REDO_RRUS_PREINT	PREINT
		WRITE_ACIA	GS
REDO_RRU3_GS	GS	REDO_RRUS_DCMOW	DCMOW
		REDO_RRUS_PREINT	PREINT
		WRITE_ACIA	GS
REDO_RRU4_GS	GS	REDO_RRUS_DCMOW	DCMOW
		REDO_RRUS_PREINT	PREINT
		WRITE_ACIA	GS
REDO_RRUS_DCMOW	DCMOW	USEREXEC	DCEXEC
		CCOW_INTP_DCMOW	DCMOW

Module Name	Located in	Is Called by	Located in
REDO_RRUS_PREINT	PREINT	USEREXEC	PREALT
		CCOW_INTP_PREINT	PREINT
REINIT_DELAY	DCMOW	MASTER_FRAME	DCMOW
RESET_ACIAS_GS	GS	INIT_DCEXEC	DCEXEC
		INIT	PREALT
RESET_BLACK_GS	GS	ENCRYPT_CCOW	DCEXEC
		DECRYPT_CCOW	DCEXEC
		DECRYPT_RCOW_GS	GS
		ENCRYPT_RCOW_GS	GS
		KG_INITIAL_GS	GS
		DECRYPT_CCOW	PREALT
		ENCRYPT_CCOW	PREALT
RESET_RRU_PTR_GS	GS	RRU_INTRPRT	DCMOW
		CALL_REQUEST	DCMOW
		DATA_XFER	DCMOW
		GLIST_DISP	DCMOW
		CALL_COMPLETE	DCMOW
		INFO_REPORT	DCMOW
		RRU_INTRPRT	PREINT
		RRU_STATUS_RPTB	PREINT
		DATA_XFER	PREINT
		GLIST_DISPLAY	PREINT
		GLIST_DISPLAY_05	PREINT
		CALL_COMPLETE	PREINT
		RRU_PRTY_OUT	PREINT
		INFO_REPORT	PREINT
		INFO_REPORT_05	PREINT
		CALL_REQUEST	PREINT
		CALL_REQ_05	PREINT
		CONF_REQUEST	PREINT
		RRU_CALL_CANCEL	PREINT
		RRU_CALLCAN_05	PREINT
		GLIST_ENTR_DEL	PREINT
		GLIST_ENTDEL_05	PREINT
		GLIST_ENTDEL_10	PREINT
		PAGING_RCOW	PREINT
		PAGING_05	PREINT
		CONF_REQ_05	PREINT
		CONFIG_ENTRY	PREINT
RGRXEX	STPV06	EXEC	STPV06
ROMTST	DBPV4	PWRON	DBPV4
RPT_B_OUT_PREINT	PREINT	FP_RCOW	PREALT
		STATUS_RPT_B	PREINT
RQST_CODE_GS	GS	INFO_REPORT	DCMOW
		FPANEL	FP
		INFO_REPORT_FP	FP
		INFO_REQUEST_GS	GS
		INFO_RPT_05	PREINT

Module Name	Located in	Is Called by	Located in
RRU_ACPT_ON_GS	GS	DATA_XFER	DCMOW
		CALL_REQUEST	DCMOW
		CALL_COMPLETE	DCMOW
		INFO_REPORT	DCMOW
		RRU_STATUS_RPTB	PREINT
		DATA_XFER	PREINT
		CALL_COMPLETE	PREINT
		RRU_PRTY_OUT_05	PREINT
		INFO_RPT_05	PREINT
		RRU_CALLCAN_15	PREINT
		PAGING_05	PREINT
		CALL_REQ_05	PREINT
		CONF_REQ_05	PREINT
RRU_B2_LOOP_CHK	DCEXEC/PREALT	RCOW_PREP_XMIT	DCEXEC
RRU_CLEAR_BIT_GS	GS	RCOW_PREP_XMIT	PREALT
		STP_MSG_INTRPRT	DCEXEC
		CLR_RRU_PREC_DSP	DCEXEC
		FPANEL	FP
		CLR_RRU_PREC_DSP	PREALT
		STP_MSG_INTRPRT	PREALT
RRU_DISCO	PREINT	MASTER_FRAME	PREINT
		CHK_USER_NR1	PREINT
RRU_INTRPRT	DCMOW/PREINT	CHK_USER_NR2	PREINT
		RRU_MSG_IN_DCMOW	DCMOW
		RRUMSG_IN_PREINT	PREINT
RRU_MSG_IN_DCMOW	DCMOW	USERPREAC	DCEXEC
RRU_MSG_UPDATE	PREALT	USEREXEC	DCEXEC
		USERPREAC	DCEXEC
		USEREXEC	PREALT
RRU_OUT_ADR	PREINT	USERPREAC	PREALT
		BUSY_ACK	PREINT
		CALL_ACK	PREINT
RRU_SET_BIT_GS	GS	STP_MSG_INTRPRT	DCEXEC
		RRU_MSG_UPDATE	DCEXEC
		FPANEL	FP
		STP_MSG_INTRPRT	PREALT
		RRU_MSG_UPDATE	PREALT
		MASTER_FRAME	PREINT
RRU_TURNOFF	PRECLA	TIME_CHK_PRECLA	PRECLA
RRU_XMIT_SET_GS	GS	USEREXEC	DCEXEC
		USERPREAC	DCEXEC
		USEREXEC	PREALT
		USERPREAC	PREALT
RRUMSG_IN_PREINT	PREINT	USEREXEC	PREALT
		USERPREAC	PREALT
RRUOUTAD2_PREINT	PREINT	FRAME_CHK_PRECLA	PRECLA
		BUSY_ACK	PREINT
		CALL-ACK_05	PREINT
		CALL_ACK	PREINT
SAVE_STP11_GS	GS	DISCO_SLOT_CHK	DCMOW
		INIT	PREALT
		RRU_TURNOFF	PRECLA
		SLOT_CON_UPDATE	PREINT
		PORT_DISCONNECT	PREINT

Module Name	Located in	Is Called by	Located in
SAVE_STP12_GS	GS	SWITCH_REDO	DCEXEC
		AC_HOME_FREQ_FP	FP
		EXEC_PREALT	PREALT
		USEREXEC	PREALT
		EXAM_CCOW_OVRHED	PRECLA
		HOME_CHAN_05	PREINT
SAVE_STP13_GS	GS	TIME_CHK_DCCNTR	DCCNTR
		EXEC_DCEXEC	DCEXEC
		USEREXEC	DCEXEC
		STP_MSG_INTRPRT	DCEXEC
		MASTER_FRAME	DCMOW
		SETUP	PREALT
		USERPREAC	PREALT
		STP_MSG_INTRPRT	PREALT
		EXAM_CCOW_OVRHED	PRECLA
		STATUS_PRECLA	PRECLA
		CC_EXEC_PRECLA	PRECLA
SAVE_STP18_GS	GS	TIME_CHK_DCCNTR	DCCNTR
		STATUS_DCCNTR	DCCNTR
		USEREXEC	DCEXEC
		USERPREAC	DCEXEC
		STP_MSG_INTRPRT	DCEXEC
		MASTER_FRAME	DCMOW
		RCOW_XMIT_DCMOW	DCMOW
		USEREXEC	PREALT
		STP_MSG_INTRPRT	PREALT
		EXAM_CCOW_OVRHED	PRECLA
		TIME_CHK_PRECLA	PRECLA
		STATUS_PRECLA	PRECLA
		RCOW_OUT_PREINT	PREINT
		MASTER_FRAME	PREINT
		LINK_TEST_ASSIGN	PREINT
SAVE_SW3_GS	GS	SETUP_MISC	DCEXEC
		AC_HOME_FREQ_FP	FP
		INIT	PREALT
		SETUP	PREALT
		EXAM_CCOW_OVRHED	PRECLA
		HOME_CHAN_05	PREINT
SCCNV	STPV06	EXEC	STPV06
SEL_PORT_DISCO	PREINT	SLOT_DISCONNECT	PREINT
		RRU_DISCO	PREINT
		SLOT_CON_UPDATE	PREINT
		CALL_COMPLETE	PREINT
SELECT_PORT_FP	GP	JUMP_SELECT_1_GS	GS
		JUMP_SELECT_2_GS	GS
SEND_CCOW_BLOCK	PRECLA	CC_EXEC_PRECLA	PRECLA
SEND_RCOW_TO_SAC	PRECLA	CC_EXEC_PRECLA	PRECLA
SEQUENCE_LITES	DCCNTR	FRAME_CHK_DCCNTR	DCCNTR
SET_DBPSTAT_MSG	DCCNTR/PRECLA	STATUS_DCCNTR	DCCNTR
		STATUS_PRECLA	PRECLA
SET_GLIST_FLAG	PREINT	GLIST_ENTDEL_10	PREINT
		GLIST_ENTDEL_30	PREINT
SET_GLIST_PNTR	PREINT	RPT_GLIST1	PREINT
		RPT_GLIST2	PREINT
		RPT_GLIST3	PREINT
		RPT_GLIST4	PREINT

Module Name	Located in	Is Called by	Located in
SET_MSG_CODE	PREINT	RPT_GLIST1	PREINT
		RPT_GLIST2	PREINT
		RPT_GLIST3	PREINT
		RPT_GLIST4	PREINT
SET_PORT_CC	DCCNTR/PRECLA	STATUS_DCCNTR	DCCNTR
		STATUS_PRECLA	PRECLA
SET_PREMPT_FLAG	PREINT	RRU_DISCO	PREINT
		SLOT_CON_UPDATE	PREINT
SET_SC_FLAG	PREINT	SET_XMIT_FLAG	PREINT
SET_TIME_CODE	PREINT	RRU_DISCO	PREINT
		SLOT_CON_UPDATE	PREINT
SETUP	DBPV4/STPV06	INIT	PREALT
SETUP_MISC	DCEXEC/PREALT	INIT_DCEXEC	DCEXEC
SETUP1	DBPV4/DCMOW	PWRON	DBPV4
		TDMA	DBPV4
SHIFT_EVEN_PORT	PREINT	STATUS_RPT_A	PREINT
SHIFT_ODD_PORT	PREINT	STATUS_RPT_A	PREINT
SHIFTLEFT	GS	LOAD_KG_VARS_GS	GS
SHIFTRIGHT	GS	LOAD_KG_VARS_GS	GS
SHOW_PORT_GS	GS	PORT_DETECT_GS	GS
SHOW_WX_GS	GS	KG_TIME_TO_DSPLY	FP
		DISP_VWXYZ_GS	GS
		BCD_TO_YZ_FP	FP
		AC_HOME_FREQ_FP	FP
SHOW_YZ_GS	GS	KG_TIME_TO_DSPLY	FP
		DISP_VWXYZ_GS	GS
		DISP_INFO_REQ_GS	GS
		FPANEL	FP
SHOWOUT_YZ_GS	GS	CONFIG_CODE_FP	FP
		CONFIG_CODE_FP	FP
SLOT_CON_UPDATE	PREINT	SLOT_CONNECT	PREINT
SOFA	DBPV4	SOF	DBPV4
SOFAA	DBPV4	RBIT	DBPV4
SRX	STPV06	CONS	STPV06
SST	STPV06	RCWI	STPV06
		RR2	STPV06
STATUS_DCCNTR	DCCNTR	EXEC_DCEXEC	DCEXEC
STATUS_PRECLA	PRECLA	EXEC_PREALT	PREALT
STC	STPV06	CONFIG_CODE_FP	STPV06
STC4	STPV06	STC	STPV06
STG_LITE_UPDATE	FP	STG_CONTROL_FP	FP
STP_MSG_INTRPT	DCEXEC/PREALT	USEREXEC	DCEXEC
		USERPREAC	DCEXEC
		USEREXEC	PREALT
		USERPREAC	PREALT
STRAP_05	GS	STRAP_READ_GS	GS
STRAP_READ_GS	GS	INIT_DCEXEC	DCEXEC
		SETUP_MISC	DCEXEC
		INIT	PREALT
STUFF	DBPV4	TDMA	DBPV4
STX	STPV06	CONS	STPV06
SUB_KG_TIME_GS	GS	PORT_DETECT_GS	GS
SWAPLITES_DCEXEC	DCEXEC	EXEC_DCEXEC	DCEXEC
		USEREXEC	DCEXEC
		STP_MSG_INTRPT	DCEXEC
		MASTER_FRAME	DCMOW
		CHAN_CONTROL_FP	FP

Module Name	Located in	Is Called by	Located in
SWITCH_REDO	DCEXEC	EXEC_DCEXEC	DCEXEC
SWITCHIT_GS	GS	EXEC_DCEXEC	DCEXEC
		SWITCH_REDO	DCEXEC
		EXEC_PREALT	PREALT
TEST_ACIA	GS	INTERUPT_GS	GS
TIME_CHK_DCCNTR	DCCNTR	USEREXEC	DCEXEC
		USERPREAC	DCEXEC
TIME_CHK_PRECLA	PRECLA	USEREXEC	PREALT
		USERPREAC	PREALT
TIME_DELAY	DCMOW/PREINT	INFO_REPORT	DCMOW
		RRU_STATUS_RPTB	PREINT
		PAGING_05	PREINT
		INFO_RPT_05	PREINT
		CALL_COMPLETE	PREINT
TIME_FMT	PREINT	RRU_DISCO	PREINT
		SLOT_CON_UPDATE	PREINT
		CALL_QUE_UPDATE	PREINT
		PORT_DISCO_UPDATE	PREINT
TS_PREP_DCEXEC	DCEXEC	FRAME_CHK_DCCNTR	DCCNTR
		USERPREAC	DCEXEC
TS_PREP_PREALT	PREALT	USERPREAC	PREALT
		FRAME_CHK_PRECLA	PRECLA
TST_XMITD_QUE	PREINT	BUSY_ACK	PREINT
		READ_CALL_ACK	PREINT
UNPACK_GS	GS	FPANEL	FP
		INFO_REQ_CHK_FP	FP
		FREQ_TO_DSPLY_FP	FP
		CONFIG_CODE_FP	FP
		SHOW_YZ_GS	GS
		SHOW_WX_GS	GS
USER1_SETUP_GS	GS	ZEROIZE	DCMOW
		INFO_REQUEST_GS	GS
		SLOT_DISCONNECT	PREINT
		PARTY_OUT_OF_SVC	PREINT
		USER1_SETUP_GS	PREINT
		ZEROIZE	PREINT
		CCOW_DATA_XFER	PREINT
		CALL_IN_QUE	PREINT
		CALL_WAITING	PREINT
		DELET_GUARD_LIST	PREINT
		ENTER_GUARD_LIST	PREINT
		HOME_CHAN_CHANG	PREINT
		CALL_CANCEL	PREINT
		FORMAT_CHANGE	PREINT
		SLOT_CONNECT	PREINT
		LINK_TEST_ASSIGN	PREINT

Module Name	Located in	Is Called by	Located in
USER2_SETUP	PREINT	ZEROIZE	DCMOW
		SLOT_DISCONNECT	PREINT
		SLOT_CONNECT	PREINT
		FORMAT_CHANGE	PREINT
		CALL_CANCEL	PREINT
		HOME_CHAN_CHANG	PREINT
		ENTER_GUARD_LIST	PREINT
		DELET_GUARD_LIST	PREINT
		CALL_IN_QUE	PREINT
		ZEROIZE	PREINT
		PARTY_OUT_OF_SVC	PREINT
		CALL_CANCEL	PREINT
		HOME_CHAN_CHANG	PREINT
USER3_SETUP	PREINT	ENTER_GUARD_LIST	PREINT
		DELET_GUARD_LIST	PREINT
		EXEC_DCEXEC	DCEXEC
USEREXEC	DCEXEC/PREALT	EXEC_PREALT	PREALT
USERPREAC	DCEXEC/PREALT	EXEC_DCEXEC	DCEXEC
		EXEC_PREALT	PREALT
VWXYZ_TO_BIN_FP	FP	ZEROIZE_FP	FP
		GLIST_DELETE_FP	FP
		PAGING_REQ_FP	FP
		GLIST_ENTER_FP	FP
		INFO_REPORT_FP	FP
		TO_PARITY_CALL_FP	FP
		CONFER_CALL_FP	FP
		TEST_ACIA	GS
WRITE_ACIA	GS	KG_TIME_FP	FP
WX_TO_BCD_FP	FP	INFO_REQUEST_FP	FP
XFER_CCOW_TO_DBP	DCEXEC	EXEC_DCEXEC	DCEXEC
XFER_OW	PREALT	EXEC_PREALT	PREALT
XFER_RCOW_DBP_GS	GS	EXEC_DCEXEC	DCEXEC
		RCOW_PREP_XMIT	DCEXEC
		RCOW_PREP_XMIT	PREALT
XPLUSA_GS	GS	FRAME_CHK_DCCNTR	DCCNTR
		CALL_REQUEST	DCCNTR
		SEQUENCE LITES	DCCNTR
		RCOW_INTP_DCCNTR	DCCNTR
		RRU_B2_LOOP_CHK	DCEXEC
		SWAPLITES_DCEXEC	DCEXEC
		CALL_REQUEST	DCMOW
		INFO_REQ_CHK_FP	FP
		INIT	PREALT
		RRU_B2_LOOP_CHK	PREALT
		RCOW_INTP_PRECLA	PRECLA
		FORMAT_GLIST_RCW	PREINT
		MASTER_FRAME	PREINT
		RRU_DISCO	PREINT
		CHK_GLIST	PREINT
		XPLUSB_GS	GS
		RRU_CLEAR_BIT_GS	GS
		RRU_SET_BIT_GS	GS

Module Name	Located in	Is Called by	Located in
XPLUSB_GS	GS	FORMAT_FREQ_FP	FP
		DC_HOME_FREQ_FP	FP
		FREQ_TO_DSPLY_FP	FP
		CALL_COMPLETE	DCMOW
		CALL_REQUEST	DCMOW
		BIT_RATE_FIND	DCMOW
		OUT_OF_SERV_FP	FP
YZ_TO_BCD_FP	FP	DC_HOME_FREQ_FP	FP
		KG_TIME_FP	FP
		INFO_REQUEST_FP	FP
		CONFER_CALL_FP	FP
YZ_TO_BIN_FP	FP		
ZERO_SLOTS_GS	GS	STATUS_DCCNTR	DCCNTR
		ZEROIZE	DCMOW

Module Name	Located in	Is Called by	Located in
ZEROFILL_GS	GS	SPECIAL_CCOW	DCCNTR
		TIME_CHK_DCCNTR	DCCNTR
		STATUS_DCCNTR	DCCNTR
		EXEC_DCEXEC	DCEXEC
		STP_MSG_INTRPRT	DCEXEC
		RCOW_XMIT_DCMOW	DCMOW
		CHAN_CONTROL_FP	FP
		EXEC_PREALT	PREALT
		STP_MSG_INTRPRT	PREALT
		TIME_CHK_PRECLA	PRECLA
		STATUS_PRECLA	PRECLA
		MASTER_FRAME	PREINT
		CONF_REQ_05	PREINT
		REDO_RRU1_GS	GS
		CLEAR_SLOTS_GS	GS

APPENDIX C

A. PARSER.ADA

```
with text_io; use text_io;
procedure parser is

  package int_io is new integer_io(integer);
  use int_io;

  subtype label_range is integer range 1..6;
  subtype opcode_range is integer range 8..13;
  subtype operand_range is integer range 15..20;

  subtype str6 is string(1..6);
  subtype str120 is string(1..120);

  type call_node;
  type call_node_ptr is access call_node;

  type call_node is record
    caller : integer;
    next   : call_node_ptr;
  end record;

  type label_node;
  type label_node_ptr is access label_node;

  type label_node is record
    label      : str6;
    called_by  : call_node_ptr;
    line       : integer;
    next       : label_node_ptr;
  end record;

  label_list : label_node_ptr := null;
  new_label  : label_node_ptr;
  search     : label_node_ptr;
  label      : str6;
  opcode     : str6;
  operand    : str6;
  col_nbr    : natural;
  new_caller : call_node_ptr;
  input_file : file_type;
  output_file : file_type;

  procedure establish_label_list is
    line      : str120;
    line_nbr  : integer := 0;
    count     : natural;
    debug     : boolean := false;
    found     : boolean;
  begin
    while not end_of_file(input_file) loop
```

```

get_line(input_file, line, count);
line_nbr := line_nbr + 1;
put(output_file, line_nbr, 4);
put_line(output_file, " " & line(1..count));
-- CHECK IF THE LINE HAS A LABEL
if count > 0 then
  if line(1) >= 'A' and line(1) <= 'Z' then
    if count >= 7 then
      label := line(label_range);
      if debug then put_line("LABEL IS " & label); end if;

      -- CHECK AND SEE IF THE LABEL EXISTS
      search := label_list;
      found := false;
      while search /= null loop
        if search.label = label then
          found := true;
        end if;
        search := search.next;
      end loop;
      if debug then put_line("SEARCHED FOR LABEL - " &
        boolean'image(found)); end if;
      if not found then
        -- ENTER THE LABEL INTO THE END OF THE LINKED LIST
        new_label := new_label_node;
        new_label.label := label;
        new_label.next := null;
        new_label.line := line_nbr;
        search := label_list;
        if search = null then
          -- FIRST ITEM IN THE LIST
          label_list := new_label;
        else
          while search.next /= null loop
            search := search.next;
          end loop;
          search.next := new_label;
        end if;
      end if;
    end if; -- count > 7
  end if; -- 'A' or 'Z'
end if; -- count > 0
end loop;
end establish_label_list;

procedure parse_control is
  line      : str120;
  count     : natural;
  line_count : integer := 0;
  debug     : boolean := false;
  found     : boolean;
  caller    : call_node_ptr;

  function control_opcode(opcode : str6) return boolean is
    debug : boolean := false;
  begin
    if debug then put(opcode); end if;

```

```

    if opcode = "BCC"    " or
      opcode = "BCS"    " or
      opcode = "BEQ"    " or
      opcode = "BGE"    " or
      opcode = "BGT"    " or
      opcode = "BHI"    " or
      opcode = "BLE"    " or
      opcode = "BLS"    " or
      opcode = "BLT"    " or
      opcode = "BMI"    " or
      opcode = "BNE"    " or
      opcode = "BPL"    " or
      opcode = "BRA"    " or
      opcode = "BSR"    " or
      opcode = "BVC"    " or
      opcode = "BVS"    " or
      opcode = "JMP"    " or
      opcode = "JSR"    " then
        if debug then put_line(" - TRUE"); end if;
        if debug then skip_line; end if;
        return true;
      else
        if debug then put_line(" - FALSE"); end if;
        if debug then skip_line; end if;
        return false;
      end if;
    end control_opcode;

begin
  while not end_of_file(input_file) loop
    get_line(input_file, line, count);
    for i in count + 1 .. line'last loop
      line(i) := ' ';
    end loop;

    line_count := line_count + 1;
    if debug then put_line(line(1..count)); end if;
    -- CHECK IF THE LINE ISN'T A COMMENT
    if line(1) /= '*' then
      -- NOT A COMMENT

      opcode := line(opcode_range);
      operand := line(operand_range);

      if control_opcode(opcode) then
        if debug then put_line("FOUND BRANCH OPCODE - LOOKING FOR " &
operand); end if;
        -- SEARCH THE LABEL LIST FOR THE OPERAND
        search := label_list;
        found := false;
        while not found loop
          --if debug then put_line("COMPARING " & operand & " TO " &
search.label); end if;
          if operand = search.label then
            -- FOUND MATCH, ADD CURRENT LINE NUMBER TO LIST
            if debug then put_line("FOUND MATCHING LABEL - ADDING
LINE NUMBER"); end if;

```

```

-- ADD TO END
new_caller      := new call_node;
new_caller.next := null;
new_caller.caller := line_count;
if search.called_by = null then
    search.called_by := new_caller;
else
    caller := search.called_by;
    while caller.next /= null loop
        caller := caller.next;
    end loop;
    caller.next := new_caller;
end if;
found := true;
else
    search := search.next;
    if search = null then
        put_line("COULD NOT FIND LABEL " & operand & " AT LINE
" & integer'image(line_count));
        found := true;
    end if;
end if;
end loop;
end if;
end if;
if debug then skip_line; end if;
end loop;
end parse_control;

procedure report_results is
begin
    while label_list /= null loop
        put(label_list.label & " (");
        if label_list.line < 10 then
            put(label_list.line, 1);
        elsif label_list.line < 100 then
            put(label_list.line, 2);
        elsif label_list.line < 1000 then
            put(label_list.line, 3);
        else
            put(label_list.line, 4);
        end if;
        put(") ");
        while label_list.called_by /= null loop
            put(" ");
            put(label_list.called_by.caller, 4);
            label_list.called_by := label_list.called_by.next;
        end loop;
        label_list := label_list.next;
        new_line;
    end loop;
end report_results;

begin
    open(input_file, in_file, "s_dbp.src");
    create(output_file, out_file, "s_dbp.lst");
    establish_label_list;

```

```
close(input_file);  
close(output_file);  
open(input_file, in_file, "s_dbp.src");  
parse_control;  
close(input_file);  
report_results;  
end parser;
```


APPENDIX D

A. DATA BUFFER PROCESSOR SOURCE CODE LISTING

```

1 *****
2 *
3 *      BEGIN POWER ON INIT
4 *
5      ORG      0DC00H
6 PWRON  LDS      #00FFFH
7      SEI
8      LDX      #IRS
9      STX      0FFF8H
10     STAA     MPUB
11     JSR      INIT
12     STAA     BRATE1    SET BIT RATES = 75 BPS
13     STAA     BRATE2
14     LDAA     #055H      **TEST RAM**
15     CLRB
16 RAMT   LDX      #0FFCH
17 RAMTST STAA     0,X
18     CMPA     0,X
19     BNE      RAMERR
20     STAA     MPUB
21     DEX
22     BNE      RAMTST
23     TSTB
24     BNE      TSTROM
25     LDAA     #0AAH      TEST OTHER RAM BITS
26     INCB
27     BRA      RAMT
28 TSTROM JSR      ROMTST  **NOP FOR TESTING**
29     JSR      CLEAR
30     BRA      PWRON1
31 RAMERR LDAA     #020H      **RAM ERROR**
32     STAA     LATCH2      *****
33     STAA     MPUB      CRASH
34     BRA      RAMERR      *****
35 *****
36 *
37 *
38 ROMTST CLRA          **TEST ROM**
39     CLRB
40     LDX      #PWRON
41 ROMTSA ADDA     0,X
42     ADCB     #0
43     STAA     MPUB
44     INX
45     CPX      #PGMTOP
46     BNE      ROMTSA
47     CMPA     CKSMA
48     BNE      ROMERR
49     CMPB     CKSMB

```

```

50      BNE      ROMERR
51      RTS
52 *
53 ROMERR LDAA   #040H      **ROM ERROR**
54      STAA    LATCH2     *****
55      STAA    MPUB       CRASH
56      BRA     ROMERR     *****
57 *
58 PWRON1 JSR     SETUP1
59      LDAA    PIA1AD      CLEAR LATCHED INPUTS
60      LDAA    PIA1BD
61      LDAA    #09DH       ENABLE INTERRUPTS ON ACIA
62      STAA    ACIACS
63      CLI
64      JMP     TDMA
65 *****
66 *
67 *
68 INIT  LDAA    #01FH      MASTER RESET ACIA
69      STAA    ACIACS      DISABLE INTERRUPTS
70      LDAA    #0FH
71      STAA    LATCH3      RESET I/O FIFOS
72      CLRA      I/O PORT & PIA INIT
73      STAA    PIA1AC
74      STAA    PIA1BC
75      STAA    IO1+1
76      STAA    IO1+3
77      STAA    IO2+1
78      STAA    IO2+3
79      STAA    IO3+1
80      STAA    IO3+3
81      STAA    IO4+1
82      STAA    IO4+3
83      STAA    IO1        SET DATA DIRECTION REG
84      STAA    IO2        A - SIDE = INPUTS
85      STAA    IO3        FOR I/O PORT
86      STAA    IO4
87      COMA
88      STAA    PIA1AD
89      STAA    PIA1BD
90      STAA    IO1+2      B-SIDE = OUTPUTS
91      STAA    IO2+2      FOR I/O PORT
92      STAA    IO3+2
93      STAA    IO4+2
94      LDAA    #03EH      SET CONTROL REG
95      STAA    PIA1AC
96      LDAA    #02EH
97      STAA    IO1+1
98      STAA    IO1+3
99      STAA    IO2+1
100     STAA    IO2+3
101     STAA    IO3+1
102     STAA    IO3+3
103     STAA    IO4+1
104     STAA    IO4+3
105     STAA    PIA1BC
106     LDAA    PIA1AD      CLEAR LATCHED INPUTS

```



```

107      LDAA    PIA1BD
108      CLRA
109      STAA    PIA1AD
110      STAA    LATCH1    CLEAR SIGNAL ACQ. LINES
111      STAA    LATCH2
112      RTS
113 *****
114 *
115 *
116 CLEAR  CLRA          CLEAR RAM MEMORY
117      LDX     #0FF8H
118 CLEARA STAA    0,X
119      DEX
120      BNE     CLEARA
121      STAA    INPT1
122      RTS
123 *****
124 *
125 *
126 SETUP  BSR     CLEAR
127 SETUP1 INCA
128      STAA    TDMAFG    SETUP RCCOW & CCOW PORTS
129      INCA
130      STAA    MSDO1
131      STAA    MSDO2
132      STAA    MSDO3
133      STAA    MSDO4
134      LDAA    #0FH      INIT FIFO MR INFO
135      STAA    MRFTA
136      STAA    LATCH3
137      LDX     #ERCRX
138      STX     RCRXPT
139      LDAA    DATA1    CHECK FOR 'TEST 1' MODE
140      BITA    #010H
141      BEQ     SETUP3
142      LDAA    #0FH
143      STAA    PTONRX
144      LDAA    #024H
145      STAA    BRATE1    INIT BIT RATES
146      STAA    BRATE2    FOR TEST #1 MODE:
147      LDAA    #020H      ALL = 2400 BPS
148      STAA    CHRAT1
149      STAA    CHRAT2
150      STAA    CHRAT3
151      STAA    CHRAT4
152 SETUP3 RTS
153 *****
154 *
155 *
156 TDMA   STAA    MPUB     ***BITE OUTPUT***
157      LDAA    DATA1    CHECK TDMA SWITCH FLAG
158      BITA    #020H      OF T&C PROCESSOR
159      BNE     TDMA1
160      LDAB    TDMAFG
161      BNE     TDMAI
162 *****
163 *

```

```

164 *NON-TDMA SERVICE
165 *
166     BITA    #01H      CHECK NON-TDMA TX REQ
167     BEQ     TDMAA
168     LDAA    #01FH      SET TX ACK
169     BRA     TDMAB
170 TDMAA LDAA    #0FH      CLEAR TX ACK
171 TDMAB STAA    LATCH3
172     BRA     TDMA
173 *
174 TDMAI SEI
175     JSR     INIT       INIT. FOR NON-TDMA
176     CLR     TDMAFG
177     CLR     PIA1BD
178     LDAA    DATA3     RESET TX FIFO
179     LDAA    #036H
180     STAA    PIA1AC     STROBE "NON-TDMA START"
181     LDAA    #03EH
182     STAA    PIA1AC
183     BRA     TDMA
184 *
185 TDMA1 LDAA    TDMAFG
186     BNE     RBI
187     STAA    PTNRX      TURN ALL PORTS OFF
188     BSR     SETUP1     INIT FOR TDMA
189     LDAA    #09DH      ENABLE RX INTERRUPTS
190     STAA    ACIACS
191     CLI
192 *****
193 *
194 *
195 RBI    LDAA    PIA1AC   CHECK RCV SETUP RDY
196     BMI     RBIA
197     JMP     TBI
198 RBIA   LDAB    SIEN
199     BEQ     RBIB
200     LDAB    BITELT      SYST INPUT FIFO BITE
201     ORAB    #020H      (RX FIFO)
202     STAB    BITELT
203     CLR     SIEN       CLEAR INPUT FIFO FLAGS
204     CLR     SELFTR
205     CLR     FILLFG
206     CLR     INPTRC
207 *
208 ***** BEGIN RECEIVE BURST INIT *****
209 *
210 RBIB   LDAB    PIA1AD   CLR LATCHED STROBE
211     LDAA    DATA4     LOAD RCV SETUP DATA
212     LDAB    SELFTR
213     BEQ     RBIBA
214     LDAB    BITELT      *SELF TEST FAIL BITE*
215     ORAB    #080H
216     STAB    BITELT
217 RBIBA  ANDA    #0C7H
218     TAB
219     ANDA    #7
220     CLR     MASKIO

```

```

221      CMPA    #1
222      BEQ     RBI1C
223      JMP     RBI2B
224 *
225 RBI1C LDAA    XMTRQ1    **PORT #1 RX BURST INIT**
226      BEQ     RBI1CA
227      JMP     TBI
228 RBI1CA LDAA    PTONRX
229      ANDA    #1
230      BNE     RBI1D
231      JMP     TBI
232 RBI1D ORAB     CHRAT1
233      LDAA    UT1
234      BEQ     RBI1E
235      LDAA    DATA6      CHECK FIFO'S "O.R."
236      BITA    #8
237      BNE     RBI1DA
238      LDAA    BITELT      ***I/O FIFO #1 BITE***
239      ORAA    #2
240      STAA    BITELT
241 RBI1DA CLR     UT1
242 RBI1E LDAA     #1
243      STAA    MASKIO      INIT I/O ON/OFF MASK
244      TSTB
245      BMI     RBI1G
246      LDAA    MSD01
247      BEQ     RBI1EA
248      CLR     FLBFG1      TWO OR MORE MISSING DO'S
249      LDAA    #2
250      STAA    MSD01
251      JMP     TBI
252 RBI1EA INCA     **FIRST MISSING DO SETUP**
253      STAA    FMSD01
254      STAA    MSD01
255      STAA    FILLFG      SETUP TO STUFF
256      LDAA    CHCT1      "FF" INTO BUFFER
257      LDAB    CHCT1+1
258      ADDB    #32          ADJ COUNT FOR I/O FIFO
259      ADCA    #0
260      STAA    MDOCT1
261      STAB    MDOCT1+1
262      LDAB    CHRAT1
263      BRA     RBI1H1
264 RBI1G LDAA     MSD01
265      CMPA    #2          TWO OR MORE MISSING DO'S?
266      BNE     RBI1H
267      LDAA    MRFTA
268      ORAA    #1          RESET I/O #1 FIFO
269      STAA    MRFTA
270      STAA    LATCH3
271 RBI1GA LDX     #SBUFF1    INIT BUFFER #1 AND
272      STX     OUTPT1      I/O FIFO #1
273      STAA    FLBFG1
274      CLRA
275      STAA    CHCT1
276      LDAA    #0FFH      **RX ELAST BUFF CH**
277      TST     BPS16K      16KBPS?

```

```

278      BEQ      RBI1GB
279      JSR      STUFF      16 KBPS ELASTIC BUF. CHARS.
280      STX      INPT1
281      LDX      #256      17 MIN. ELASTIC BUFF
282      STX      CHCT1
283      BRA      RBI1H
284 RBI1GB STAA      SBUFF1      RCV BURST
285      STAA      SBUFF1+1
286      STAA      SBUFF1+2
287      STAA      SBUFF1+3
288      STAA      SBUFF1+4
289      LDAA      #5
290      STAA      CHCT1+1
291      LDX      #SBUFF1+5
292      STX      INPT1
293 RBI1H  CLR      MSDO1
294 RBI1H1 LDX      #INPT1      INIT FOR INPUT
295      STX      RXPNTR      FIFO SERVICE
296      LDAA      BPS16K
297      BEQ      RBI1HA      16 KBPS?
298      LDX      #EBUFF4
299      STX      RXUVAL      INIT 16 KBPS BUFFER PNTRS.
300      LDX      #SBUFF1
301      STX      RXLVAL
302      BRA      RBI1HB
303 RBI1HA LDX      #EBUFF1
304      STX      RXUVAL
305      LDX      #SBUFF1
306      STX      RXLVAL
307 RBI1HB LDAA      SIGACQ      SET "SIGNAL ACQ. I/O #1"
308      ORAA      #1
309      STAA      SIGACQ
310      STAA      LATCH1
311      JMP      RBIS
312 *
313 RBI2B  CMPA      #2
314      BEQ      RBI2C
315      JMP      RBI3B
316 RBI2C  LDAA      XMTRQ2      **PORT #2 BURST INIT**
317      BEQ      RBI2CA
318      JMP      TBI
319 RBI2CA LDAA      PTONRX
320      ANDA      #2
321      BNE      RBI2D
322      JMP      TBI
323 RBI2D  ORAB      CHRAT2
324      LDAA      UT2
325      BEQ      RBI2E
326      LDAA      DATA6      CHECK FIFO'S "O.R."
327      BITA      #4
328      BNE      RBI2DA
329      LDAA      BITELT      ***I/O FIFO #2 BITE***
330      ORAA      #4
331      STAA      BITELT
332 RBI2DA CLR      UT2
333 RBI2E  LDAA      #2
334      STAA      MASKIO      INIT I/O ON/OFF MASK

```

```

335      TSTB
336      BMI      RBI2G
337      LDAA      MSDO2
338      BEQ      RBI2EA
339      CLR      FLBFG2      TWO OR MORE MISSING DO'S
340      LDAA      #2
341      STAA      MSDO2
342      JMP      TBI
343 RBI2EA INCA      **FIRST MISSING DO SETUP**
344      STAA      FMSDO2
345      STAA      MSDO2
346      STAA      FILLFG      SETUP TO STUFF
347      LDAA      CHCT2      "FF" INTO BUFFER
348      LDAB      CHCT2+1
349      ADDB      #32      ADJ COUNT FOR I/O FIFO
350      ADCA      #0
351      STAA      MDOCT2
352      STAB      MDOCT2+1
353      LDAB      CHRAT2
354      BRA      RBI2H1
355 RBI2G LDAA      MSDO2
356      CMPA      #2      TWO OR MORE MISSING DO'S?
357      BNE      RBI2H
358      LDAA      MRFTA
359      ORAA      #2      RESET I/O #2 FIFO
360      STAA      MRFTA
361      STAA      LATCH3
362 RBI2GA LDX      #SBUFF2      INIT BUFFER #2 AND
363      STX      OUTPT2      I/O FIFO #2
364      STAA      FLBFG2
365      CLRA
366      STAA      CHCT2
367      LDAA      #0FFH      **RX ELAST BUFF CH**
368      TST      BPS16K      16KBPS?
369      BEQ      RBI2GB
370      JSR      STUFF      16KBPS ELASTIC BUF. CHARS.
371      STX      INPT2
372      LDX      #256      17 MIN ELAST BUFF
373      STX      CHCT2
374      LDX      #SBUFF1
375      STX      OUTPT2
376      BRA      RBI2H
377 RBI2GB STAA      SBUFF2      RCV BURST
378      STAA      SBUFF2+1
379      STAA      SBUFF2+2
380      STAA      SBUFF2+3
381      STAA      SBUFF2+4
382      LDAA      #5
383      STAA      CHCT2+1
384      LDX      #SBUFF2+5
385      STX      INPT2
386 RBI2H CLR      MSDO2
387 RBI2H1 LDX      #INPT2      INIT FOR INPUT
388      STX      RXPNTR      FIFO SERVICE
389      LDAA      BPS16K
390      BEQ      RBI2HA      16 KBPS?
391      LDX      #EBUFF4

```

```

392      STX      RXUVAL      INIT. 16KBPS BUF. PNTRS.
393      LDX      #SBUFF1
394      STX      RXLVAL
395      BRA      RBI2HB
396 RBI2HA LDX      #EBUFF2
397      STX      RXUVAL
398      LDX      #SBUFF2
399      STX      RXLVAL
400 RBI2HB LDAA     SIGACQ      SET "SIGNAL ACQ. I/O #2"
401      ORAA     #2
402      STAA     SIGACQ
403      STAA     LATCH1
404      JMP      RBIS
405 *
406 RBI3B  CMPA     #3
407      BEQ      RBI3C
408      JMP      RBI4B
409 RBI3C  LDAA     XMTRQ3      **PORT #3 BURST INIT**
410      BEQ      RBI3CA
411      JMP      TBI
412 RBI3CA LDAA     PTONRX
413      ANDA     #4
414      BNE      RBI3D
415      JMP      TBI
416 RBI3D  ORAB     CHRAT3
417      LDAA     UT3
418      BEQ      RBI3E
419      LDAA     DATA6      CHECK FIFO'S "O.R."
420      BITA     #2
421      BNE      RBI3DA
422      LDAA     BITELT      **I/O FIFO #3 BITE***
423      ORAA     #8
424      STAA     BITELT
425 RBI3DA CLR      UT3
426 RBI3E  LDAA     #4
427      STAA     MASKIO      INIT I/O ON/OFF MASK
428      TSTB
429      BMI      RBI3G
430      LDAA     MSDO3
431      BEQ      RBI3EA
432      CLR      FLBFG3
433      LDAA     #2
434      STAA     MSDO3
435      JMP      TBI
436 RBI3EA INCA     **FIRST MISSING DO SETUP**
437      STAA     FMSDO3
438      STAA     MSDO3
439      STAA     FILLFG      SETUP TO STUFF
440      LDAA     CHCT3      "FF" INTO BUFFER
441      LDAB     CHCT3+1
442      ADDB     #32      ADJ COUNT FOR I/O FIFO
443      ADCA     #0
444      STAA     MDOCT3
445      STAB     MDOCT3+1
446      LDAB     CHRAT3
447      BRA      RBI3H1
448 RBI3G  LDAA     MSDO3

```

```

449      CMPA    #2          TWO OR MORE MISSING DO'S?
450      BNE     RBI3H
451      LDAA    MRFTA
452      ORAA    #4          RESET I/O FIFO #3
453      STAA    MRFTA
454      STAA    LATCH3
455 RBI3GA LDX     #SBUFF3    INIT BUFFER #3 AND
456      STX      OUTPT3      I/O FIFO #3
457      STAA    FLBFG3
458      CLRA
459      STAA    CHCT3
460      LDAA    #0FFH        **RX ELAST BUFF CH**
461      TST     BPS16K        16KBPS?
462      BEQ     RBI3GB
463      JSR     STUFF         16 KBPS ELASTIC BUF. CHARS.
464      STX     INPT3
465      LDX     #256          17 MIN ELASTIC BUFF.
466      STX     CHCT3
467      LDX     #SBUFF1
468      STX     OUTPT3
469      BRA     RBI3H
470 RBI3GB STAA    SBUFF3      RCV BURST
471      STAA    SBUFF3+1
472      STAA    SBUFF3+2
473      STAA    SBUFF3+3
474      STAA    SBUFF3+4
475      LDAA    #5
476      STAA    CHCT3+1
477      LDX     #SBUFF3+5
478      STX     INPT3
479 RBI3H  CLR     MSD03
480 RBI3H1 LDX     #INPT3      INIT FOR INPUT
481      STX     RXPNTR        FIFO SERVICE
482      LDAA    BPS16K
483      BEQ     RBI3HA
484      LDX     #EBUFF4
485      STX     RXUVAL        INIT 16KBPS PNTRS
486      LDX     #SBUFF1
487      STX     RXLVAL
488      BRA     RBI3HB
489 RBI3HA LDX     #EBUFF3
490      STX     RXUVAL
491      LDX     #SBUFF3
492      STX     RXLVAL
493 RBI3HB LDAA    SIGACQ      SET "SIGNAL ACQ. I/O #3"
494      ORAA    #4
495      STAA    SIGACQ
496      STAA    LATCH1
497      JMP     RBIS
498 *
499 RBI4B  CMPA    #4
500      BEQ     RBI4C
501      JMP     RBIJ
502 RBI4C  LDAA    XMTRQ4      **PORT #4 BURST INIT**
503      BEQ     RBI4CA
504      JMP     TBI
505 RBI4CA LDAA    PTONRX

```

506	ANDA	#8	
507	BNE	RBI4D	
508	JMP	TBI	
509	RBI4D	ORAB	CHRT4
510	LDAA	UT4	
511	BEQ	RBI4E	
512	LDAA	DATA6	CHECK FIFO'S "O.R."
513	BITA	#1	
514	BNE	RBI4DA	
515	LDAA	BITELT	**I/O FIFO #4 BITE**
516	ORAA	#010H	
517	STAA	BITELT	
518	RBI4DA	CLR	UT4
519	RBI4E	LDAA	#8
520	STAA	MASKIO	INIT I/O ON/OFF MASK
521	TSTB		
522	BMI	RBI4G	
523	LDAA	MSDO4	
524	BEQ	RBI4EA	
525	CLR	FLBFG4	TWO OR MORE MISSING DO'S
526	LDAA	#2	
527	STAA	MSDO4	
528	JMP	TBI	
529	RBI4EA	INCA	**FIRST MISSING DO SETUP**
530	STAA	FMSDO4	
531	STAA	MSDO4	
532	STAA	FILLFG	SETUP TO STUFF
533	LDAA	CHCT4	"FF" INTO BUFFER
534	LDAB	CHCT4+1	
535	ADDB	#32	ADJ COUNT FOR I/O FIFO
536	ADCA	#0	
537	STAA	MDOCT4	
538	STAB	MDOCT4+1	
539	LDAB	CHRT4	
540	BRA	RBI4H1	
541	RBI4G	LDAA	MSDO4
542	CMPA	#2	TWO OR MORE MISSING DO'S?
543	BNE	RBI4H	
544	LDAA	MRFTA	
545	ORAA	#08H	RESET I/O FIFO #4
546	STAA	MRFTA	
547	STAA	LATCH3	
548	RBI4GA	LDX	#SBUFF4 INIT BUFFER #4 AND
549	STX	OUTPT4	I/O FIFO #4
550	STAA	FLBFG4	
551	CLRA		
552	STAA	CHCT4	
553	LDAA	#0FFH	**RX ELAST BUFF CH**
554	TST	BPS16K	16KBPS?
555	BEQ	RBI4GB	
556	JSR	STUFF	16 KBPS ELASTIC BUFF. CHARS
557	STX	INPT4	
558	LDX	#256	17 MIN ELASTIC BUFF
559	STX	CHCT4	
560	LDX	#SBUFF1	
561	STX	OUTPT4	
562	BRA	RBI4H	


```

563 RBI4GB STAA SBUFF4 RCV BURST
564 STAA SBUFF4+1
565 STAA SBUFF4+2
566 STAA SBUFF4+3
567 STAA SBUFF4+4
568 LDAA #5
569 STAA CHCT4+1
570 LDX #SBUFF4+5
571 STX INPT4
572 RBI4H CLR MSDO4
573 RBI4H1 LDX #INPT4 INIT FOR INPUT
574 STX RXPNTR FIFO SERVICE
575 LDAA BPS16K
576 BEQ RBI4HA
577 LDX #EBUFF4
578 STX RXUVAL INIT 16KBPS PNTRS
579 LDX #SBUFF1
580 STX RXLVAL
581 BRA RBI4HB
582 RBI4HA LDX #EBUFF4
583 STX RXUVAL
584 LDX #SBUFF4
585 STX RXLVAL
586 RBI4HB LDAA SIGACQ SET "SIGNAL ACQ I/O #4"
587 ORAA #8
588 STAA SIGACQ
589 STAA LATCH1
590 JMP RBIS
591 *****
592 *
593 *
594 RBIJ CMPA #7 CHECK FOR SELF TEST RX BURS
595 BNE RBIJA
596 STAA SELFTR
597 LDAA #0 INIT. RX FIFO COUNT
598 LDAB #60
599 JMP RBIT
600 RBIJA LDX RCRXPT
601 CPX #ERCRX R/CCOW RX BUFFER EMPTY?
602 BEQ RBIK
603 PSHA **RCCOW/CCOW ERROR**
604 LDAA BITELT
605 ORAA #1
606 STAA BITELT
607 PULA
608 LDX #ERCRX
609 STX RCRXPT
610 RBIK LDX #SRCRX
611 CMPA #6
612 BNE RBIN
613 TSTB CCOW DO?
614 BMI RBIL
615 LDAA #0A5H OUTPUT CCOW DO
616 RBIKA STAA ACIAD ABSENT CODE TO SCU
617 JMP TBI
618 *
619 RBIL LDAA DATA1 TX REQUEST STATUS WBR

```

```

620      ANDA    #0FH                                WBR
621      ORAA    #0C0H                                WBR
622      STAA    ACIAD    SEND C CODE TO SCP          WBR
623  *
624      LDAA    #0A1H    CCOW DO PRESENT CODE
625  RBIM    STAA    0,X
626      INX
627      STX      RCRXPT    ADJ R/CCOW RX BUFFER PNTR
628      STAA    INPTRC
629      STAA    SIEN
630      JMP      TBI
631  RBIN    TSTB      RCCOW DO?
632      BMI      RBIO
633      LDAA    #0A6H    RCCOW DO ABSENT CODE
634      BRA      RBIKA
635  RBIO    LDAA    #0A2H    RCCOW DO PRESENT CODE
636      BRA      RBIM
637  RBIS    BSR      BITRAT
638      BRA      RBIT
639  *****
640  *
641  *
642  BITRAT  TST      BPS16K    TEST 16KBPS
643          BNE      RBISF
644          ASLB
645          BMI      RBIU
646          ANDB    #070H
647          BNE      RBISA
648          LDAB    #13        75 BPS @ 1.386 SEC
649          CLRA          13 BYTES
650          RTS
651  RBISA   CMPB    #010H
652          BNE      RBISB
653          LDAB    #52        300 BPS @ 1.386 SEC
654          CLRA          52 BYTES
655          RTS
656  RBISB   CMPB    #020H
657          BNE      RBISC
658          LDAB    #104       600 BPS @ 1.386 SEC
659          CLRA          104 BYTES
660          RTS
661  RBISC   CMPB    #030H
662          BNE      RBISD
663          LDAB    #208       1200 BPS @ 1.386 SEC
664          CLRA          208 BYTES
665          RTS
666  RBISD   CMPB    #040H
667          BNE      RBISE
668          LDAB    #0A0H       2400 BPS @ 1.386 SEC
669          LDAA    #1          416 BYTES
670          RTS
671  RBISE   LDAB    #040H       4800 BPS @ 1.386 SEC
672          LDAA    #3          832 BYTES
673          RTS
674  RBISF   LDAB    #0D5H       16000 BPS @ 1.386 SEC
675          LDAA    #0AH       2773 BYTES
676          RTS

```

```

677 RBIU   LDAB   #78      75 BPS @ 8.32 SEC
678       CLRA           78 BYTES
679       RTS
680 *****
681 *
682 *
683 RBIT     STAA   INCHC    MSB'S OF CH. CNT.
684       STAB   INCHC+1    LSB'S OF CH. CNT.
685       CLRA
686       STAA   INPTRC
687       INCA
688       STAA   SIEN
689 *****
690 *
691 *
692 TBI      LDAA   CDRDY     TEST CCOW DATA RDY FLG.
693       BEQ     TBI1
694       CLR     CDRDY
695       LDAA   PA1AD
696       ORAA   #020H      SET DATA RDY CCOW*
697       BRA     TBI2
698 TBI1     LDAA   RDRDY     TEST RCCOW RDY FLG.
699       BEQ     TBI3
700       CLR     RDRDY
701       LDAA   PA1AD
702       ORAA   #010H      SET DATA RDY RCCOW*
703 TBI2     STAA   PA1AD
704       STAA   PIA1AD
705 TBI3     LDAA   IRSBIT    IRS BITE?
706       BEQ     TBI4
707       LDAA   BITELT
708       ORAA   #01H
709       STAA   BITELT
710 TBI4     LDAA   PIA1BC    CHECK TX SETUP READY
711       BMI     TBIA
712       JMP     IO1PT
713 TBIA     LDAA   SOEN
714       BEQ     TBIB
715       LDAB   BITELT      SYST OUTPUT FIFO BITE
716       ORAB   #040H      (TX FIFO)
717       STAB   BITELT
718 *
719 ***** BEGIN TRANSMIT BURST INITIALIZATION *****
720 *
721 TBIB     LDAA   PIA1BD    CLR LATCHED STROBE
722       LDAA   DATA3      LOAD TX SETUP DATA
723       LDAB   DATA5      CHECK TX INHIBIT
724       BITB   #1
725       BEQ     TIBA
726       ANDA   #7
727       CMPA   #7
728       BEQ     TBIBA
729       JMP     IO1PT      TX INHIBIT: ABORT
730 TBIBA    JMP     TBIG     ALLOW SELF TEST BURST
731 TIBA     ANDA   #047H
732       TAB
733       ANDA   #7

```

```

734      CMPA    #1
735      BNE     TBI2B
736      LDAA    PTONRX      IS PORT ON?
737      BITA    #1
738      BNE     TBI1BA
739      JMP     IO1PT      ABORT:  PORT IS OFF
740 *
741 TBI1BA TST     BPS16K
742      BEQ     TBI1E
743      LDX     #SBUFF1      16 KBPS
744      STX     TXLVAL
745      LDX     #EBUFF4
746      STX     TXUVAL
747      BRA     TBI1EA
748 TBI1E LDX     #SBUFF1      INIT FOR OUTPUT
749      STX     TXLVAL      FIFO SERVICE
750      LDX     #EBUFF1
751      STX     TXUVAL
752 TBI1EA LDX     #INPT1
753      STX     TXPNTR
754      CLRA
755      STAA    UT1
756      STAA    MDOCT1
757      STAA    MDOCT1+1
758      STAA    FMSDO1
759      LDAA    #2
760      STAA    MSDO1
761      LDAA    #0FEH
762      STAA    XMASK
763      LDAA    #0EFH
764      STAA    MASKR
765      LDAA    GROUP1
766      BEQ     TBI1F
767      STAA    GRP1FG
768 TBI1F CLRA
769      STAA    GROUP1
770      ORAB    CHRAT1
771      JMP     TBIN
772 *
773 TBI2B CMPA    #2
774      BNE     TBI3B
775      LDAA    PTONRX      IS PORT ON?
776      BITA    #2
777      BNE     TBI2BA
778      JMP     IO1PT      ABORT:  PORT IS OFF
779 TBI2BA TST     BPS16K
780      BEQ     TBI2E
781      LDX     #SBUFF1      16 KBPS
782      STX     TXLVAL
783      LDX     #EBUFF4
784      STX     TXUVAL
785      BRA     TBI2EA
786 TBI2E LDX     #SBUFF2      INIT. FOR OUTPUT
787      STX     TXLVAL      FIFO SERVICE
788      LDX     #EBUFF2
789      STX     TXUVAL
790 TBI2EA LDX     #INPT2

```

791	STX	TXPNTR	
792	CLRA		
793	STAA	UT2	
794	STAA	MDOCT2	
795	STAA	MDOCT2+1	
796	STAA	FMSDO2	
797	LDAA	#2	
798	STAA	MSDO2	
799	LDAA	#0FDH	
800	STAA	XMASK	
801	LDAA	#0DFH	
802	STAA	MASKR	
803	LDAA	GROUP2	
804	BEQ	TBI2F	
805	STAA	GRP1FG	
806	TBI2F	CLRA	
807	STAA	GROUP2	
808	ORAB	CHRTAT2	
809	JMP	TBIN	
810	*		
811	TBI3B	CMPA	#3
812	BNE	TBI4B	
813	LDAA	PTONRX	IS PORT ON?
814	BITA	#4	
815	BNE	TBI3BA	
816	JMP	IO1PT	ABORT: PORT IS OFF
817	TBI3BA	TST	BPS16K
818	BEQ	TBI3E	
819	LDX	#SBUFF1	16 KBPS
820	STX	TXLVAL	
821	LDX	#EBUFF4	
822	STX	TXUVAL	
823	BRA	TBI3EA	
824	TBI3E	LDX	#SBUFF3 INIT. FOR OUPUT
825	STX	TXLVAL	FIFO SERVICE
826	LDX	#EBUFF3	
827	STX	TXUVAL	
828	TBI3EA	LDX	#INPT3
829	STX	TXPNTR	
830	CLRA		
831	STAA	UT3	
832	STAA	MDOCT3	
833	STAA	MDOCT3+1	
834	STAA	FMSDO3	
835	LDAA	#2	
836	STAA	MSDO3	
837	LDAA	#0FBH	
838	STAA	XMASK	
839	LDAA	#0BFH	
840	STAA	MASKR	
841	LDAA	GROUP3	
842	BEQ	TBI3F	
843	STAA	GRP1FG	
844	TBI3F	CLRA	
845	STAA	GROUP3	
846	ORAB	CHRTAT3	
847	JMP	TBIN	

```

848 *
849 TBI4B  CMPA  #4
850          BNE  TBIG
851          LDAA PTONRX      IS PORT ON?
852          BITA  #8
853          BNE  TBI4BA
854          JMP  IO1PT      ABORT:  PORT IS OFF
855 TBI4BA  TST  BPS16K
856          BEQ  TBI4E
857          LDX  #SBUFF1    16 Kbps
858          STX  TXLVAL
859          LDX  #EBUFF4
860          STX  TXUVAL
861          BRA  TBI4EA
862 TBI4E  LDX  #SBUFF4      INIT. FOR OUTPUT
863          STX  TXLVAL      FIFO SERVICE
864          LDX  #EBUFF4
865          STX  TXUVAL
866 TBI4EA  LDX  #INPT4
867          STX  TXPNTR
868          CLRA
869          STAA UT4
870          STAA MDOCT4
871          STAA MDOCT4+1
872          STAA FMDO4
873          LDAA #2
874          STAA MSDO4
875          LDAA #0F7H
876          STAA XMASK
877          LDAA #07FH
878          STAA MASKR
879          LDAA GROUP4
880          BEQ  TBI4F
881          STAA GRP1FG
882 TBI4F  CLRA
883          STAA GROUP4
884          ORAB  CHRAT4
885          JMP  TBIN
886 *
887 TBIG  CMPA  #7          CHECK FOR SELF TEST TX BURS
888          BNE  TBIGA
889          STAA SOEN
890          STAA SELFTT
891          LDX  #08000H    DUMMY VALUE FOR TXPNTR
892          STX  TXPNTR
893          LDX  #28          SELF TEST CH CNT - 32
894          STX  OUTCHC
895          LDAA #32
896          LDAB #033H      SELF TEST TX CHAR
897 TBIG1  STAB  PIA1BD
898          DECA          OUTPUT 32 SELF TEST
899          BNE  TBIG1      CHARS. TO TX FIFO
900          JMP  RBI
901 TBIGA  CMPA  #6
902          BNE  TBIJ
903          LDAA PA1AD      IF NO DATA READY, OUTPUT
904          BITA  #020H      13 BYTES OF 11001100

```

```

905      BNE      TBIGC
906      LDAA     #0CCH      **NO CCOW DATA RDY**
907      LDX      #13
908 TBIGB STAA     PIA1BD
909      DEX
910      BNE      TBIGB
911      BRA      TBIHB
912 TBIGC LDX      #SCTXB
913      STX      CTXPT
914 TBIH  LDAA     0,X
915      STAA     PIA1BD      PUT CHAR IN OUTPUT FIFO
916      INX
917      CPX      #ECTXB
918      BNE      TBIH
919      LDAA     PA1AD
920      ANDA     #0DFH      CLR DATA READY CCOW
921 TBIHA STAA     PA1AD
922      STAA     PA1AD
923 *
924 *
925 TBIHB ASLB
926      BMI      TBIHC
927      JMP      IO1PT
928 TBIHC CLRA
929      STAA     PIA1BD
930      STAA     PIA1BD
931      STAA     PIA1BD
932      JMP      IO1PT
933 TBIJ  LDX      #SRTXB
934      STX      RTXPT
935 TBIK  LDAA     0,X
936      STAA     PIA1BD      PUT CHAR IN OUTPUT FIFO
937      INX
938      CPX      #ERTXB
939      BNE      TBIK
940      LDAA     PA1AD
941      ANDA     #0EFH      CLR DATA READY RCCOW
942      BRA      TBIHA
943 TBIN  JSR      BITRAT      GET TX CH. CNT.
944 TBIO  STAA     OUTCHC      MSB'S OF CH. CNT.
945      STAB     OUTCHC+1     LSB'S OF CH. CNT.
946      CLRA
947      INCA
948      STAA     SOEN
949 TBIQ  LDX      TXPNTR
950      LDAA     GRP1FG      CHECK IF FIRST GROUP
951      BEQ      TBIS
952 TBIQ1 LDAA     OUTCHC      ***COMPUTE FILL COUNT***
953      LDAB     OUTCHC+1
954      ADDB     #6          ADD ELASTIC BUFF(48 BITS)
955      ADCA     #0
956      SUBB     5,X          BURST CH CNT + ELASTIC BUFF
957      SBCA     4,X          - BUFF CH CNT = FILL COUNT
958      BCS      TBIR
959      BNE      TBIRA
960      TSTB
961      BNE      TBIRA

```

```

962 TBIR    CLR    GRPIFG
963         BRA    TBIS      NO FILL REQUIRED
964 TBIRA    STAA   FILLCT
965         STAB   FILLCT+1
966         LDAA   4,X        NUMBER OF GOOD CHARS IN FIR
967         LDAB   5,X        BURST=BUFF CH CNT - ELASTIC
968         SUBB   #6        SUBT ELASTIC BUFF(48 BITS)
969         SBCA   #0
970         BCC    TBIRB
971         CLRA                   ERROR: ELASTIC BUFFER NOT
972         STAA   SOEN          YET FULL FOR TRANSMISSION
973         LDAA   BITELT
974         ORAA   #040H        **TX FIFO BITE LITE ON**
975         STAA   BITELT
976         BRA    IO1PT
977 TBIRB    STAA   OUTCHC
978         STAB   OUTCHC+1
979 TBIS     JSR    SOFAA        *SERVICE OUTPUT FIFO*
980 *****
981 *
982 *
983 *
984 IO1PT    LDAA   PTONRX      PORT #1 ON?
985         ANDA   #1
986         BEQ    IO1A1
987         CLR    P1OFF
988         JMP    IO1A
989 IO1A1    LDAA   P1OFF      PORT IS OFF
990         BEQ    IO1A2
991         JMP    IO2PT
992 IO1A2    STAA   CHCT1
993         STAA   CHCT1+1
994         STAA   XMTRQ1
995         STAA   UT1
996         LDX    #0
997         STX    LGDPT1
998         STAA   OUFQ1
999         STAA   EMPTY1
1000        LDAA   #2
1001        STAA   MSDO1
1002        INCA
1003        STAA   FLBFG1
1004        LDAA   MRFTA
1005        ORAA   #1          RESET FIFO I/O #1
1006        ANDA   #0FEH      CLR "XMT ACK"
1007        STAA   MRFTA
1008        STAA   LATCH3
1009        STAA   P1OFF
1010        LDAA   PA1AD
1011        ANDA   #0FEH      CLEAR "DATA RDY PORT #1"
1012        STAA   PA1AD
1013        STAA   PIA1AD
1014        LDX    #SBUFF1
1015        STX    INPT1
1016        STX    OUTPT1
1017        LDAA   SIGACQ      CLR "SIGNAL ACQ. I/O #1"
1018        ANDA   #0FEH

```



```

1019      STAA    SIGACQ
1020      STAA    LATCH1
1021      LDX     TXPNTR
1022      CPX     #INPT1
1023      BNE     IO1A3
1024      CLR     SOEN
1025 IO1A3  JMP     IO2PT
1026 *****
1027 *
1028 *   INPUT SERVICE SUBROUTINE
1029 *
1030 IO1LS  LDAA    #1
1031      STAA    XFER1
1032      LDAA    IO1      OBTAIN CHAR
1033      LDX     INPT1    GET PUT PNTR
1034      STAA    0,X      PUT IT
1035      INX                      INCR PUT PNTR
1036      CPX     OUTPT1   TEST OVERFLOW
1037      BNE     IO1LSX   BRANCH NO OVERFLOW
1038      LDAA    MRFTA    **OVERFLOW**
1039      ANDA    #0EFH    CLR "XMT ACK"
1040      ORAA    #1      RESET FIFO
1041      STAA    MRFTA
1042      STAA    LATCH3
1043      STAA    OUFG1    SET OUFG FLAG
1044      INS                      *REMOVE RETURN
1045      INS                      *
1046      JMP     IO2PT
1047 IO1LSX LDAA    BPS16K
1048      BEQ     IO1LSA
1049      CPX     #EBUFF4   16 KBPS
1050      BNE     IO1LSB
1051      LDX     #SBUFF1
1052      BRA     IO1LSB
1053 IO1LSA CPX     #EBUFF1   TEST END OF INPUT
1054      BNE     IO1LSB    NO
1055      LDX     #SBUFF1   GET INITIAL VALUE
1056 IO1LSB STX     INPT1
1057      LDX     CHCT1    GET CHAR COUNT
1058      INX                      INCR COUNT
1059      STX     CHCT1    PUT BACK
1060      RTS
1061 *****
1062 *
1063 *LOAD I/O #1 FIFO WITH 1 CHAR. ROUTINE
1064 *
1065 IO1LF  LDX     OUTPT1
1066      LDAA    RXINHB
1067      BITA    #1      IS THE EXTEND TRANSMIT FLAG SET?
1068      BNE     IO1LF1   YES:RX NOT ALLOWED
1069      LDAA    0,X
1070      STAA    IO1+2    OUTPUT CH. TO I/O #1 FIFO
1071 IO1LF1 INX
1072      LDAB    BPS16K
1073      BEQ     IO1LFA
1074      CPX     #EBUFF4   16 KBPS
1075      BNE     IO1LFB

```

```

1076      LDX      #SBUFF1
1077      BRA      IO1LFB
1078 IO1LFA CPX      #EBUFF1
1079      BNE      IO1LFB
1080      LDX      #SBUFF1
1081 IO1LFB STX      OUTPT1
1082      LDAA     #1
1083      STAA     XFER1
1084      LDAA     FMSDO1    TEST MISSING DO
1085      BEQ      IO1LFC
1086      LDX      MDOCT1
1087      DEX
1088      STX      MDOCT1
1089      BNE      IO1LFC
1090      LDAA     SIGACQ
1091      ANDA     #0FEH
1092      STAA     SIGACQ
1093      STAA     LATCH1
1094      CLR      FMSDO1
1095 IO1LFC LDX      CHCT1    ADJUST CHAR COUNT
1096      DEX
1097      STX      CHCT1
1098      BNE      IO1LFD
1099      INS
1100      INS
1101      JMP      IO1YE1    GO CHECK UNDERFLOW
1102 IO1LFD RTS
1103 *****
1104 *
1105 *LOAD I/O #1 FIFO WITH 3 BUF. CH. ROUTINE
1106 *
1107 LD1    LDX      OUTPT1
1108      LDAA     0,X
1109      STAA     IO1+2
1110      LDAA     1,X
1111      STAA     IO1+2
1112      LDAA     2,X
1113      STAA     IO1+2
1114      INX
1115      INX
1116      INX
1117      STX      OUTPT1
1118      LDX      CHCT1
1119      DEX
1120      DEX
1121      DEX
1122      STX      CHCT1
1123      RTS
1124 *****
1125 *
1126 ***** BEGIN SERVICING I/O PORT #1 *****
1127 *
1128 IO1A   LDAA     PTONRX    RX ONLY?
1129      BITA     #010H
1130      BEQ      IO1AA
1131      JMP      IO1Y
1132 IO1AA  LDAA     DATA1

```

1133	BITA	#1	
1134	BNE	IO1B	
1135	JMP	IO1Y	
1136	IO1B	ASLA	
1137	BPL	IO1C	
1138	LDAA	XMTRQ1	
1139	BNE	IO1C	
1140	LDAA	EMPTY1	
1141	BNE	IO1CA	
1142	LDX	CHCT1	R/T SELECTED: CHECK RCV DON
1143	BEQ	IO1BB	
1144	IO1BA	JMP	IO1YD
1145	IO1BB	LDAA	DATA6
1146	BITA	#8	TEST: I/O FIFO UNDERFLOW
1147	BNE	IO1BA	BRANCH: NO UNDERFLOW
1148	IO1C	LDAA	XMTRQ1
1149	BNE	IO1D	
1150	LDAA	EMPTY1	
1151	BEQ	IO1E	
1152	IO1CA	CLR	EMPTY1
1153	STAA	XMTRQ1	GET HERE W/MULTIPLE
1154	LDX	#0	TX REQ PER FRAME
1155	STX	LGDP1	
1156	LDAA	MRFTA	
1157	ORAA	#010H	SET "TX ACK"
1158	STAA	MRFTA	
1159	STAA	LATCH3	
1160	IO1D	LDAA	OUFFG1
1161	BEQ	IO1H	
1162	STAA	XFER1	
1163	JMP	IO2PT	
1164	IO1E	LDAA	DATA5
1165	ANDA	#1	CHECK TX INHIBIT
1166	BEQ	IO1F	
1167	JMP	IO2PT	
1168	IO1F	LDAA	#1
1169	STAA	XMTRQ1	**INIT FOR PORT #1**
1170	STAA	GROUP1	**DATA TRANSMISSION**
1171	STAA	XFER1	
1172	STAA	FLBFG1	
1173	LDAA	SIGACQ	
1174	ORAA	#010H	SELECT I/O INPUT MODE
1175	ANDA	#0FEH	CLR "SIG. ACQ. #1"
1176	STAA	SIGACQ	
1177	STAA	LATCH1	
1178	LDAA	MRFTA	
1179	ORAA	#1	TOGGLE FIFO'S RESET
1180	STAA	LATCH3	
1181	CLR	IO1+2	CLR PIA B-SIDE
1182	ANDA	#0FEH	
1183	ORAA	#010H	SET "XMT ACK" I/O #1
1184	STAA	MRFTA	
1185	STAA	LATCH3	
1186	LDX	#SBUFF1+6	
1187	STX	INPT1	
1188	LDX	#SBUFF1	
1189	STX	OUTPT1	

1190	LDAA	#0FFH	STUFF 6 ELASTIC
1191	STAA	0,X	BUFFER CHARACTERS (CC)
1192	STAA	1,X	
1193	STAA	2,X	
1194	STAA	3,X	
1195	STAA	4,X	
1196	STAA	5,X	
1197	LDX	#6	
1198	STX	CHCT1	
1199	JMP	IO2PT	
1200	IO1G	LDX	#SBUFF1
1201	STX	INPT1	
1202	STX	OUTPT1	
1203	LDX	#0	
1204	STX	CHCT1	
1205	LDAA	SIGACQ	CLR "SIGNAL ACQ. I/O #1
1206	ANDA	#0FEH	
1207	STAA	SIGACQ	
1208	STAA	LATCH1	
1209	JMP	IO2PT	
1210	IO1H	LDAA	DATA5
1211	ANDA	#1	CHECK TX INHIBIT
1212	BEQ	IO1HA	
1213	JMP	IO1YA1	
1214	IO1HA	LDAB	FLBFG1
1215	BEQ	IO1HB	
1216	JMP	IO1O	
1217	IO1HB	LDAA	DATA6
1218	BITA	#8	TEST FIFO "O.R."
1219	BNE	IO1I	
1220	JMP	IO1XP	
1221	IO1I	JSR	IO1LS
1222	LDAA	DATA6	LOAD & STORE 1 CHAR
1223	BITA	#8	
1224	BNE	IO1K	
1225	JMP	IO1XP	
1226	IO1K	JSR	IO1LS
1227	LDAA	DATA6	LOAD & STORE 2 CHAR
1228	BITA	#8	
1229	BNE	IO1M	
1230	JMP	IO1XP	
1231	IO1M	JSR	IO1LS
1232	LDAA	DATA6	LOAD & STORE 3 CHAR
1233	BITA	#8	
1234	BNE	IO1N2	
1235	JMP	IO1XP	
1236	IO1N2	JSR	IO1LS
1237	LDAA	BPS16K	LOAD & STORE 4 CHAR
1238	BNE	IO1N3	
1239	JMP	IO1XP	
1240	IO1N3	LDAA	DATA6
1241	BITA	#8	*16KBPS: ALLOW MORE*
1242	BNE	IO1N4	*I/O SERVICING*
1243	JMP	IO1XP	
1244	IO1N4	JSR	IO1LS
1245	LDAA	DATA6	LOAD & STORE 5 CHAR.
1246	BITA	#8	

1247	BNE	IO1N5	
1248	JMP	IO1XP	
1249	IO1N5	JSR	IO1LS LOAD & STORE 6 CHAR.
1250	LDAA	DATA6	
1251	BITA	#8	
1252	BNE	IO1N6	
1253	JMP	IO1XP	
1254	IO1N6	JSR	IO1LS LOAD & STORE 7 CHAR.
1255	LDAA	DATA6	
1256	BITA	#8	
1257	BNE	IO1N7	
1258	JMP	IO1XP	
1259	IO1N7	JSR	IO1LS LOAD & STORE 8 CHAR.
1260	LDAA	DATA6	
1261	BITA	#8	
1262	BNE	IO1N8	
1263	JMP	IO1XP	
1264	IO1N8	JSR	IO1LS LOAD & STORE 9 CHAR.
1265	LDAA	DATA6	
1266	BITA	#8	
1267	BNE	IO1N9	
1268	JMP	IO1XP	
1269	IO1N9	JSR	IO1LS LOAD & STORE 10 CHAR.
1270	LDAA	DATA6	
1271	BITA	#8	
1272	BNE	IO1N10	
1273	JMP	IO1XP	
1274	IO1N10	JSR	IO1LS LOAD & STORE 11 CHAR.
1275	JMP	IO1XP	
1276	IO1O	LDAA	DATA6
1277	BITA	#8	
1278	BEQ	IO1XP	
1279	IO1P	CLR	FLBFG1
1280	LDAA	PA1AD	SET "DATA RDY PORT #1"
1281	ORAA	#1	
1282	STAA	PA1AD	
1283	STAA	PIA1AD	
1284	JMP	IO1I	
1285	IO1XP	LDAA	EMPTY1 **TEST IF SETUP**
1286	BPL	IO1XQ	**LAST GOOD DATA PNTR**
1287	LDX	INPT1	DO SETUP
1288	STX	LGDPT1	
1289	LDAA	#1	
1290	STAA	EMPTY1	
1291	IO1XQ	JMP	IO2PT
1292	IO1Y	LDAA	XMTRQ1
1293	BEQ	IO1YB	
1294	LDAA	MRFTA	
1295	ANDA	#0EFH	CLEAR "TX ACK"
1296	STAA	MRFTA	
1297	STAA	LATCH3	
1298	CLR	XMTRQ1	FIRST TIME THAT
1299	LDAA	#0FFH	TX REQ = FALSE
1300	STAA	EMPTY1	
1301	LDAA	PA1AD	
1302	ANDA	#1	
1303	BNE	IO1YA	

1304	STAA	CHCT1	NO DATA: ABORT INPUT
1305	STAA	CHCT1+1	
1306	STAA	EMPTY1	
1307	INCA		
1308	STAA	FLBFG1	
1309	LDAA	MRFTA	
1310	ORAA	#1	
1311	STAA	MRFTA	
1312	STAA	LATCH3	
1313	JMP	IO2PT	
1314	IO1YA	JMP	IO1D
1315	IO1YB	LDAA	EMPTY1
1316		BNE	IO1YA
1317		JMP	IO1YD
1318	IO1YA1	CLRA	
1319		STAA	XMTRQ1
1320		STAA	EMPTY1
1321		LDAA	PA1AD
1322		ANDA	#0FEH
1323		STAA	PIA1AD CLEAR "DATA RDY PORT #1"
1324		STAA	PA1AD
1325		LDAA	MRFTA
1326		ORAA	#1 RESET I/O #1 FIFO
1327		ANDA	#0EFH CLR TX ACK I/O #1
1328		STAA	MRFTA
1329		STAA	LATCH3
1330	IO1YC	LDAA	MRFTA **INIT I/O PORT **
1331		ORAA	#1 **TO OUTPUT**
1332		STAA	MRFTA **RX BURST DATA**
1333		STAA	LATCH3 RESET I/O #1 FIFO
1334		LDAA	SIGACQ
1335		ANDA	#0EFH SELECT I/O OUTPUT MODE
1336		STAA	SIGACQ
1337		STAA	LATCH1
1338	IO1YC1	CLR	FMSDO1
1339		LDAA	#2
1340		STAA	MSDO1
1341		STAA	FLBFG1
1342		JMP	IO1G
1343	IO1YD	LDAA	FLBFG1
1344		BEQ	IO1YF
1345		STAA	XFER1
1346		LDAA	CHCT1+1
1347		CMPA	#10 RCV ELASTIC BUFFER FULL?
1348		BGE	IO1YE
1349		JMP	IO2PT
1350	IO1YE	CLR	FLBFG1
1351		STAA	UT1
1352		LDAA	SIGACQ SET OUTPUT MODE
1353		ANDA	#0EFH
1354		STAA	SIGACQ
1355		STAA	LATCH1
1356		LDAA	MRFTA
1357		ANDA	#0FEH CLR RESET ON I/O FIFO
1358		STAA	MRFTA
1359		STAA	LATCH3
1360		JSR	LD1

1361	JSR	LD1	LOAD FIFO W/6 BUF. CH.
1362	LDAA	BPS16K	
1363	BEQ	IO1YE0	
1364	JSR	LD1	16KBPS SO:
1365	JSR	LD1	
1366	JSR	LD1	
1367	JSR	LD1	
1368	JSR	LD1	LOAD FIFO WITH 21 BUF. CH.
1369	IO1YE0	JMP	IO2PT
1370	IO1YE1	LDAA	#1
1371		STAA	XFER1
1372		LDAA	UT1
1373		BNE	IO1YE0
1374		LDAA	DATA6
1375		BITA	#8
1376		BEQ	IO1YC1
1377		JMP	IO2PT
1378	IO1YF	LDX	CHCT1
1379		BEQ	IO1YE1
1380		LDAA	DATA6
1381		LDAB	UT1
1382		BNE	IO1YFA
1383		BITA	#8
1384		BEQ	IO1YC1
1385	IO1YFA	BITA	#080H
1386		BEQ	IO2PT
1387	IO1YG	JSR	IO1LF
1388		LDAA	DATA6
1389		BITA	#080H
1390		BEQ	IO2PT
1391	IO1YH	JSR	IO1LF
1392		LDAA	DATA6
1393		BITA	#080H
1394		BEQ	IO2PT
1395	IO1YI	JSR	IO1LF
1396		LDAA	BPS16K
1397		BEQ	IO2PT
1398	IO1YJ	LDAA	DATA6
1399		BITA	#080H
1400		BEQ	IO2PT
1401	IO1YK	JSR	IO1LF
1402		LDAA	DATA6
1403		BITA	#080H
1404		BEQ	IO2PT
1405	IO1YL	JSR	IO1LF
1406		LDAA	DATA6
1407		BITA	#080H
1408		BEQ	IO2PT
1409	IO1YM	JSR	IO1LF
1410		LDAA	DATA6
1411		BITA	#080H
1412		BEQ	IO2PT
1413	IO1YN	JSR	IO1LF
1414		LDAA	DATA6
1415		BITA	#080H
1416		BEQ	IO2PT
1417	IO1YO	JSR	IO1LF

```

1418      LDAA    DATA6
1419      BITA    #080H      CHECK "I.R.": FULL?
1420      BEQ     IO2PT
1421 IO1YP JSR     IO1LF      OUTPUT 9 CH. TO FIFO
1422      LDAA    DATA6
1423      BITA    #080H      CHECK "I.R.": FULL?
1424      BEQ     IO2PT
1425 IO1YQ JSR     IO1LF      OUTPUT 10 CH. TO FIFO
1426 *****
1427 *
1428 *
1429 *
1430 IO2PT LDAA    PTONRX      PORT #2 ON?
1431      ANDA    #2
1432      BEQ     IO2A1
1433      CLR     P2OFF
1434      JMP     IO2A
1435 IO2A1 LDAA    P2OFF      PORT IS OFF
1436      BEQ     IO2A2
1437      JMP     IO3PT
1438 IO2A2 STAA    CHCT2
1439      STAA    CHCT2+1
1440      STAA    XMTRQ2
1441      STAA    UT2
1442      LDX     #0
1443      STX     LGDPT2
1444      STAA    OUF2
1445      STAA    EMPTY2
1446      LDAA    #2
1447      STAA    MSDO2
1448      INCA
1449      STAA    FLBFG2
1450      LDAA    MRFTA
1451      ORAA    #2          RESET FIFO I/O #2
1452      ANDA    #0DFH
1453      STAA    MRFTA
1454      STAA    LATCH3
1455      STAA    P2OFF
1456      LDAA    PA1AD
1457      ANDA    #0FDH      CLEAR "DATA RDY PORT #2"
1458      STAA    PA1AD
1459      STAA    PIA1AD
1460      LDX     #SBUFF2
1461      STX     INPT2
1462      STX     OUTPT2
1463      LDAA    SIGACQ      CLR "SIGNAL ACQ. I/O #2"
1464      ANDA    #0FDH
1465      STAA    SIGACQ
1466      STAA    LATCH1
1467      LDX     TXPNTR
1468      CPX     #INPT2
1469      BNE     IO2A3
1470      CLR     SOEN
1471 IO2A3 JMP     IO3PT
1472 *****
1473 *
1474 *      INPUT SERVICE ROUTINE

```



```

1475 *
1476 IO2LS LDAA #1
1477 STAA XFER2
1478 LDAA IO2 OBTAIN CHAR
1479 LDX INPT2 GET PUT PNTR
1480 STAA 0,X PUT IT
1481 INX INCR PUT PNTR
1482 CPX OUTPT2 TEST OVERFLOW
1483 BNE IO2LSX BRANCH NO OVERFLOW
1484 LDAA MRFTA **OVERFLOW**
1485 ANDA #0DFH CLR "XMT ACK"
1486 ORAA #2 RESET FIFO
1487 STAA MRFTA
1488 STAA LATCH3
1489 STAA OUFG2 SET OUFG FLAG
1490 INS *REMOVE RETURN
1491 INS *
1492 JMP IO3PT
1493 IO2LSX LDAA BPS16K
1494 BEQ IO2LSA
1495 CPX #EBUFF4 16 KBPS
1496 BNE IO2LSB
1497 LDX #SBUFF1
1498 BRA IO2LSB
1499 IO2LSA CPX #EBUFF2 TEST END OF INPUT
1500 BNE IO2LSB NO
1501 LDX #SBUFF2 GET INITIAL VALUE
1502 IO2LSB STX INPT2
1503 LDX CHCT2 GET CHAR COUNT
1504 INX INCR COUNT
1505 STX CHCT2 PUT BACK
1506 RTS
1507 *****
1508 *
1509 *LOAD I/O #2 FIFO WITH 1 CHAR ROUTINE
1510 *
1511 IO2LF LDX OUTPT2
1512 LDAA RXINHBR WBR
1513 BITA #2 IS THE EXTEND TRANSMIT FLAG SET? WBR
1514 BNE IO2LF1 YES:RX NOT ALLOWED WBR
1515 LDAA 0,X
1516 STAA IO2+2 OUTPUT CH. TO I/O #2 FIFO
1517 IO2LF1 INX ADJ. OUTPUT PNTR.
1518 LDAB BPS16K
1519 BEQ IO2LFA
1520 CPX #EBUFF4 16 KBPS
1521 BNE IO2LFB
1522 LDX #SBUFF1
1523 BRA IO2LFB
1524 IO2LFA CPX #EBUFF2
1525 BNE IO2LFB
1526 LDX #SBUFF2
1527 IO2LFB STX OUTPT2
1528 LDAA #1
1529 STAA XFER2
1530 LDAA FMDO2 TEST MISSING DO
1531 BEQ IO2LFC

```

```

1532      LDX      MDOCT2
1533      DEX
1534      STX      MDOCT2
1535      BNE      IO2LFC
1536      LDAA     SIGACQ
1537      ANDA     #0FDH
1538      STAA     SIGACQ
1539      STAA     LATCH1
1540      CLR      FMSDO2
1541 IO2LFC LDX      CHCT2      ADJUST CHAR COUNT
1542      DEX
1543      STX      CHCT2
1544      BNE      IO2LFD
1545      INS
1546      INS      REMOVE RETURN
1547      JMP      IO2YE1      GO CHECK UNDERFLOW
1548 IO2LFD RTS
1549 *****
1550 *
1551 * LOAD FIFO I/O #2 WITH 3 BUF. CH. ROUTINE
1552 *
1553 LD2      LDX      OUTPT2
1554      LDAA     0,X
1555      STAA     IO2+2
1556      LDAA     1,X
1557      STAA     IO2+2
1558      LDAA     2,X
1559      STAA     IO2+2
1560      INX
1561      INX
1562      INX
1563      STX      OUTPT2
1564      LDX      CHCT2
1565      DEX
1566      DEX
1567      DEX
1568      STX      CHCT2
1569      RTS
1570 *****
1571 *
1572 ***** BEGIN SERVICING I/O PORT #2 *****
1573 *
1574 IO2A      LDAA     PTONRX      RX ONLY?
1575      BITA     #020H
1576      BEQ      IO2AA
1577      JMP      IO2Y
1578 IO2AA      LDAA     DATA1
1579      BITA     #2
1580      BNE      IO2B
1581      JMP      IO2Y
1582 IO2B      ASLA
1583      BPL      IO2C
1584      LDAA     XMTRQ2
1585      BNE      IO2C
1586      LDAA     EMPTY2
1587      BNE      IO2CA
1588      LDX      CHCT2      R/T SELECTED: CHECK RCV DON

```

```

1589      BEQ      IO2BB
1590 IO2BA  JMP      IO2YD
1591 IO2BB  LDAA     DATA6      TEST: I/O FIFO UNDERFLOW
1592      BITA     #4
1593      BNE      IO2BA      BRANCH: NO UNDERFLOW
1594 IO2C   LDAA     XMTRQ2
1595      BNE      IO2D
1596      LDAA     EMPTY2
1597      BEQ      IO2E
1598 IO2CA  CLR      EMPTY2      GET HERE W/MULTIPLE
1599      STAA     XMTRQ2      TX REQ PER FRAME
1600      LDX      #0
1601      STX      LGDPT2
1602      LDAA     MRFTA
1603      ORAA     #020H      SET "TX ACK"
1604      STAA     MRFTA
1605      STAA     LATCH3
1606 IO2D   LDAA     OUF2
1607      BNE      IO2DA
1608      JMP      IO2H
1609 IO2DA  STAA     XFER2
1610      JMP      IO3PT
1611 IO2E   LDAA     DATA5      CHECK TX INHIBIT
1612      ANDA     #1
1613      BEQ      IO2F
1614      JMP      IO3PT
1615 IO2F   LDAA     #1          **INIT FOR PORT #2**
1616      STAA     XMTRQ2      **DATA TRANSMISSION**
1617      STAA     GROUP2
1618      STAA     XFER2
1619      STAA     FLBFG2
1620      LDAA     SIGACQ
1621      ORAA     #020H      SELECT I/O INPUT MODE
1622      ANDA     #0FDH      CLR "SIG. ACQ. #2"
1623      STAA     SIGACQ
1624      STAA     LATCH1
1625      LDAA     MRFTA
1626      ORAA     #2          TOGGLE FIFO'S RESET
1627      STAA     LATCH3
1628      CLR      IO2+2      CLR PIA B-SIDE
1629      ANDA     #0FDH
1630      ORAA     #020H      SET "XMT ACK" I/O #2
1631      STAA     MRFTA
1632      STAA     LATCH3
1633      LDAA     BPS16K
1634      BEQ      IO2FA
1635      LDX      #SBUFF1+6      16KBPS
1636      STX      INPT2
1637      LDX      #SBUFF1
1638      BRA      IO2FB
1639 IO2FA  LDX      #SBUFF2+6
1640      STX      INPT2
1641      LDX      #SBUFF2
1642 IO2FB  STX      OUTPT2
1643      LDAA     #0FFH      STUFF 6 ELASTIC
1644      STAA     0,X      BUFFER CHARACTERS (CC)
1645      STAA     1,X

```

1646	STAA	2,X	
1647	STAA	3,X	
1648	STAA	4,X	
1649	STAA	5,X	
1650	LDX	#6	
1651	STX	CHCT2	
1652	JMP	IO3PT	
1653	IO2G	LDAA	BPS16K
1654		BEQ	IO2GA
1655		LDX	#SBUFF1 16KBPS
1656		BRA	IO2GB
1657	IO2GA	LDX	#SBUFF2
1658	IO2GB	STX	INPT2
1659		STX	OUTPT2
1660		LDX	#0
1661		STX	CHCT2
1662		LDAA	SIGACQ CLR "SIGNAL ACQ. I/O #2"
1663		ANDA	#0FDH
1664		STAA	SIGACQ
1665		STAA	LATCH1
1666		JMP	IO3PT
1667	IO2H	LDAA	DATA5 CHECK TX INHIBIT
1668		ANDA	#1
1669		BEQ	IO2HA
1670		JMP	IO2YA1
1671	IO2HA	LDAB	FLBFG2
1672		BEQ	IO2HB
1673		JMP	IO2O
1674	IO2HB	LDAA	DATA6 TEST FIFO "O.R."
1675		BITA	#4
1676		BNE	IO2I
1677		JMP	IO2XP
1678	IO2I	JSR	IO2LS LOAD & STORE 1 CHAR
1679		LDAA	DATA6
1680		BITA	#4
1681		BNE	IO2K
1682		JMP	IO2XP
1683	IO2K	JSR	IO2LS LOAD & STORE 2 CHAR
1684		LDAA	DATA6
1685		BITA	#4
1686		BNE	IO2M
1687		JMP	IO2XP
1688	IO2M	JSR	IO2LS LOAD & STORE 3 CHAR
1689		LDAA	DATA6
1690		BITA	#4
1691		BNE	IO2N2
1692		JMP	IO2XP
1693	IO2N2	JSR	IO2LS LOAD & STORE 4 CHAR
1694		LDAA	BPS16K
1695		BNE	IO2N3
1696		JMP	IO2XP
1697	IO2N3	LDAA	DATA6 *16KBPS: ALLOW MORE*
1698		BITA	#4 *I/O SERVICING*
1699		BNE	IO2N4
1700		JMP	IO2XP
1701	IO2N4	JSR	IO2LS LOAD & STORE 5 CHAR.
1702		LDAA	DATA6

1703	BITA	#4	
1704	BNE	IO2N5	
1705	JMP	IO2XP	
1706	IO2N5 JSR	IO2LS	LOAD & STORE 6 CHAR.
1707	LDAA	DATA6	
1708	BITA	#4	
1709	BNE	IO2N6	
1710	JMP	IO2XP	
1711	IO2N6 JSR	IO2LS	LOAD & STORE 7 CHAR.
1712	LDAA	DATA6	
1713	BITA	#4	
1714	BNE	IO2N7	
1715	JMP	IO2XP	
1716	IO2N7 JSR	IO2LS	LOAD & STORE 8 CHAR.
1717	LDAA	DATA6	
1718	BITA	#4	
1719	BNE	IO2N8	
1720	JMP	IO2XP	
1721	IO2N8 JSR	IO2LS	LOAD & STORE 9 CHAR.
1722	LDAA	DATA6	
1723	BITA	#4	
1724	BNE	IO2N9	
1725	JMP	IO2XP	
1726	IO2N9 JSR	IO2LS	LOAD & STORE 10 CHAR.
1727	LDAA	DATA6	
1728	BITA	#4	
1729	BNE	IO2N10	
1730	JMP	IO2XP	
1731	IO2N10 JSR	IO2LS	LOAD & STORE 11 CHAR.
1732	JMP	IO2XP	
1733	IO2O LDAA	DATA6	
1734	BITA	#4	
1735	BEQ	IO2XP	
1736	IO2P CLR	FLBFG2	
1737	LDAA	PA1AD	SET "DATA RDY PORT #2"
1738	ORAA	#2	
1739	STAA	PA1AD	
1740	STAA	PIA1AD	
1741	JMP	IO2I	
1742	IO2XP LDAA	EMPTY2	**TEST IF SETUP**
1743	BPL	IO2XQ	**LAST GOOD DATA PNTR**
1744	LDX	INPT2	DO SETUP
1745	STX	LGDP2	
1746	LDAA	#1	
1747	STAA	EMPTY2	
1748	IO2XQ JMP	IO3PT	
1749	IO2Y LDAA	XMTRQ2	
1750	BEQ	IO2YB	
1751	LDAA	MRFTA	
1752	ANDA	#0DFH	CLR "TX ACK"
1753	STAA	MRFTA	
1754	STAA	LATCH3	
1755	CLR	XMTRQ2	FIRST TIME THAT
1756	LDAA	#0FFH	TX REQ = FALSE
1757	STAA	EMPTY2	
1758	LDAA	PA1AD	
1759	ANDA	#2	

1760	BNE	IO2YA	
1761	STAA	CHCT2	NO DATA: ABORT INPUT
1762	STAA	CHCT2+1	
1763	STAA	EMPTY2	
1764	INCA		
1765	STAA	FLBFG2	
1766	LDAA	MRFTA	
1767	ORAA	#2	
1768	STAA	MRFTA	
1769	STAA	LATCH3	
1770	JMP	IO3PT	
1771	IO2YA	JMP	IO2D
1772	IO2YB	LDAA	EMPTY2
1773	BNE	IO2YA	
1774	JMP	IO2YD	
1775	IO2YA1	CLRA	
1776	STAA	XMTRQ2	
1777	STAA	EMPTY2	
1778	LDAA	PA1AD	
1779	ANDA	#0FDH	
1780	STAA	PIA1AD	CLEAR "DATA RDY PORT #2"
1781	STAA	PA1AD	
1782	LDAA	MRFTA	
1783	ORAA	#2	RESET I/O #2 FIFO
1784	ANDA	#0DFH	CLR TX ACK I/O #2
1785	STAA	MRFTA	
1786	STAA	LATCH3	
1787	IO2YC	LDAA	MRFTA **INIT I/O PORT**
1788	ORAA	#2	**TO OUTPUT**
1789	STAA	MRFTA	**RX BURST DATA**
1790	STAA	LATCH3	RESET I/O #2 FIFO
1791	LDAA	SIGACQ	
1792	ANDA	#0DFH	SELECT I/O OUTPUT MODE
1793	STAA	SIGACQ	
1794	STAA	LATCH1	
1795	IO2YC1	CLR	FMSDO2
1796	LDAA	#2	
1797	STAA	MSDO2	
1798	STAA	FLBFG2	
1799	JMP	IO2G	
1800	IO2YD	LDAA	FLBFG2
1801	BEQ	IO2YF	
1802	STAA	XFER2	
1803	LDAA	CHCT2+1	
1804	CMPA	#10	RCV ELASTIC BUFFER FULL?
1805	BGE	IO2YE	
1806	JMP	IO3PT	
1807	IO2YE	CLR	FLBFG2
1808	STAA	UT2	
1809	LDAA	SIGACQ	SET OUTPUT MODE
1810	ANDA	#0DFH	
1811	STAA	SIGACQ	
1812	STAA	LATCH1	
1813	LDAA	MRFTA	
1814	ANDA	#0FDH	CLR REST ON I/O FIFO
1815	STAA	MRFTA	
1816	STAA	LATCH3	

1817	JSR	LD2	
1818	JSR	LD2	LOAD FIFO W/6 BUF. CH.
1819	LDAA	BPS16K	
1820	BEQ	IO2YE0	
1821	JSR	LD2	16 KBPS SO:
1822	JSR	LD2	
1823	JSR	LD2	
1824	JSR	LD2	
1825	JSR	LD2	LOAD FIFO W/21 BUF. CH.
1826	IO2YE0	JMP	IO3PT
1827	IO2YE1	LDAA	#1
1828	STAA	XFER2	
1829	LDAA	UT2	
1830	BNE	IO2YE0	
1831	LDAA	DATA6	CHECK FIFO UNDERFLOW
1832	BITA	#4	BY TESTING FIFO "O.R."
1833	BEQ	IO2YC1	**UNDERFLOW**
1834	JMP	IO3PT	
1835	IO2YF	LDX	CHCT2
1836	BEQ	IO2YE1	
1837	LDAA	DATA6	
1838	LDAB	UT2	
1839	BNE	IO2YFA	
1840	BITA	#4	
1841	BEQ	IO2YC1	
1842	IO2YFA	BITA	#040H CHECK FIFO "I.R.": FULL?
1843	BEQ	IO3PT	
1844	IO2YG	JSR	IO2LF OUTPUT 1 CH. TO FIFO
1845	LDAA	DATA6	
1846	BITA	#040H	CHECK "I.R.": FULL?
1847	BEQ	IO3PT	
1848	IO2YH	JSR	IO2LF OUTPUT 2 CH. TO FIFO
1849	LDAA	DATA6	
1850	BITA	#040H	CHECK "I.R.": FULL?
1851	BEQ	IO3PT	
1852	IO2YI	JSR	IO2LF OUTPUT 3 CH. TO FIFO
1853	LDAA	BPS16K	
1854	BEQ	IO3PT	
1855	IO2YJ	LDAA	DATA6 16 KBPS: SERVICE 10 CHARS.
1856	BITA	#040H	CHECK "I.R.": FULL?
1857	BEQ	IO3PT	
1858	IO2YK	JSR	IO2LF OUTPUT 4 CH. TO FIFO
1859	LDAA	DATA6	
1860	BITA	#040H	CHECK "I.R.": FULL?
1861	BEQ	IO3PT	
1862	IO2YL	JSR	IO2LF OUTPUT 5 CH. TO FIFO
1863	LDAA	DATA6	
1864	BITA	#040H	CHECK "I.R.": FULL?
1865	BEQ	IO3PT	
1866	IO2YM	JSR	IO2LF OUTPUT 6 CH. TO FIFO
1867	LDAA	DATA6	
1868	BITA	#040H	CHECK "I.R.": FULL?
1869	BEQ	IO3PT	
1870	IO2YN	JSR	IO2LF OUTPUT 7 CH. TO FIFO
1871	LDAA	DATA6	
1872	BITA	#040H	CHECK "I.R.": FULL?
1873	BEQ	IO3PT	

```

1874 IO2YO JSR IO2LF OUTPUT 8 CH. TO FIFO
1875 LDAA DATA6
1876 BITA #040H CHECK "I.R.": FULL?
1877 BEQ IO3PT
1878 IO2YP JSR IO2LF OUTPUT 9 CH. TO FIFO
1879 LDAA DATA6
1880 BITA #040H CHECK "I.R.": FULL?
1881 BEQ IO3PT
1882 IO2YQ JSR IO2LF OUTPUT 10 CH. TO FIFO
1883 *****
1884 *
1885 *
1886 *
1887 IO3PT LDAA PTONRX PORT #3 ON?
1888 ANDA #4
1889 BEQ IO3A1
1890 CLR P3OFF
1891 JMP IO3A
1892 IO3A1 LDAA P3OFF PORT IS OFF
1893 BEQ IO3A2
1894 JMP IO4PT
1895 IO3A2 STAA CHCT3
1896 STAA CHCT3+1
1897 STAA XMTRQ3
1898 STAA UT3
1899 LDX #0
1900 STX LGDPT3
1901 STAA OUF3
1902 STAA EMPTY3
1903 LDAA #2
1904 STAA MSD03
1905 INCA
1906 STAA FLBFG3
1907 LDAA MRFTA
1908 ORAA #4 RESET FIFO I/O #3
1909 ANDA #0BFH CLR XMT ACK I/O #3
1910 STAA MRFTA
1911 STAA LATCH3
1912 STAA P3OFF
1913 LDAA PA1AD
1914 ANDA #0FBH CLEAR "DATA RDY PORT #3"
1915 STAA PA1AD
1916 STAA PIA1AD
1917 LDX #SBUFF3
1918 STX INPT3
1919 STX OUTPT3
1920 LDAA SIGACQ CLR "SIGNAL ACQ. I/O #3"
1921 ANDA #0FBH
1922 STAA SIGACQ
1923 STAA LATCH1
1924 LDX TXPNTR
1925 CPX #INPT3
1926 BNE IO3A3
1927 CLR SOEN
1928 IO3A3 JMP IO4PT
1929 *****
1930 *

```



```

1931 *   INPUT SERVICE SUBROUTINE
1932 *
1933 IO3LS  LDAA    #1
1934         STAA    XFER3
1935         LDAA    IO3      OBTAIN CHAR
1936         LDX     INPT3    GET PUT PNTR
1937         STAA    0,X      PUT IT
1938         INX
1939         CPX     OUTPT3    INCR PUT PNTR
1940         BNE     IO3LSX    TEST OVERFLOW
1941         LDAA    MRFTA     BRANCH NO OVERFLOW
1942         ANDA    #0BFH     **OVERFLOW**
1943         ORAA    #4        CLR XMT ACK I/O #3
1944         STAA    MRFTA     RESET FIFO I/O #3
1945         STAA    LATCH3
1946         STAA    OUFG3     SET OUFG FLAG
1947         INS
1948         INS              *REMOVE RETURN
1949         JMP     IO4PT     *
1950 IO3LSX LDAA    BPS16K
1951         BEQ     IO3LSA
1952         CPX     #EBUFF4   16 KBPS
1953         BNE     IO3LSB
1954         LDX     #SBUFF1
1955         BRA     IO3LSB
1956 IO3LSA CPX     #EBUFF3    TEST END OF INPUT
1957         BNE     IO3LSB    NO
1958         LDX     #SBUFF3    GET INITIAL VALUE
1959 IO3LSB STX     INPT3
1960         LDX     CHCT3     GET CHAR. COUNT
1961         INX
1962         STX     CHCT3     INCR COUNT
1963         RTS              PUT BACK
1964 *****
1965 *
1966 *LOAD I/O #3 FIFO WITH 1 CHAR. ROUTINE
1967 *
1968 IO3LF  LDX     OUTPT3
1969         LDAA    RXINHB
1970         BITA    #4        IS THE EXTEND TRANSMIT FLAG SET?
1971         BNE     IO3LF1    YES:RX NOT ALLOWED
1972         LDAA    0,X
1973         STAA    IO3+2     OUTPUT CH. TO I/O #3 FIFO
1974 IO3LF1 INX
1975         LDAB    BPS16K    ADJ. OUTPUT PNTR.
1976         BEQ     IO3LFA
1977         CPX     #EBUFF4   16KBPS
1978         BNE     IO3LFB
1979         LDX     #SBUFF1
1980         BRA     IO3LFB
1981 IO3LFA CPX     #EBUFF3
1982         BNE     IO3LFB
1983         LDX     #SBUFF3
1984 IO3LFB STX     OUTPT3
1985         LDAA    #1
1986         STAA    XFER3
1987         LDAA    FMSDO3    TEST MISSING DO

```

```

1988      BEQ      IO3LFC
1989      LDX      MDOCT3
1990      DEX
1991      STX      MDOCT3
1992      BNE      IO3LFC
1993      LDAA     SIGACQ
1994      ANDA     #0FBH
1995      STAA     SIGACQ
1996      STAA     LATCH1
1997      CLR      FMSDO3
1998 IO3LFC LDX      CHCT3      ADJ CHAR. COUNT
1999      DEX
2000      STX      CHCT3
2001      BNE      IO3LFD
2002      INS                      REMOVE RETURN
2003      INS
2004      JMP      IO3YE1      GO CHECK UNDERFLOW
2005 IO3LFD RTS
2006 *****
2007 *
2008 *LOAD FIFO I/O #3 WITH 3 BUF. CH. ROUTINE
2009 *
2010 LD3      LDX      OUTPT3
2011      LDAA     0,X
2012      STAA     IO3+2
2013      LDAA     1,X
2014      STAA     IO3+2
2015      LDAA     2,X
2016      STAA     IO3+2
2017      INX
2018      INX
2019      INX
2020      STX      OUTPT3
2021      LDX      CHCT3
2022      DEX
2023      DEX
2024      DEX
2025      STX      CHCT3
2026      RTS
2027 *****
2028 *
2029 ***** BEGIN SERVICING I/O PORT #3 *****
2030 *
2031 IO3A      LDAA     PTONRX      RX ONLY?
2032      BITA     #040H
2033      BEQ      IO3AA
2034      JMP      IO3Y
2035 IO3AA      LDAA     DATA1
2036      BITA     #4
2037      BNE      IO3B
2038      JMP      IO3Y
2039 IO3B      ASLA
2040      BPL      IO3C
2041      LDAA     XMTRQ3
2042      BNE      IO3C
2043      LDAA     EMPTY3
2044      BNE      IO3CA

```

2045	LDX	CHCT3	R/T SELECTED: CHECK RCV DON
2046	BEQ	IO3BB	
2047	IO3BA	JMP	IO3YD
2048	IO3BB	LDAA	DATA6
2049		BITA	#2
2050		BNE	IO3BA
2051	IO3C	LDAA	XMTRQ3
2052		BNE	IO3D
2053		LDAA	EMPTY3
2054		BEQ	IO3E
2055	IO3CA	CLR	EMPTY3
2056		STAA	XMTRQ3
2057		LDX	#0
2058		STX	LGDP3
2059		LDAA	MRFTA
2060		ORAA	#040H
2061		STAA	MRFTA
2062		STAA	LATCH3
2063	IO3D	LDAA	OUG3
2064		BNE	IO3DA
2065		JMP	IO3H
2066	IO3DA	STAA	XFER3
2067		JMP	IO4PT
2068	IO3E	LDAA	DATA5
2069		ANDA	#1
2070		BEQ	IO3F
2071		JMP	IO4PT
2072	IO3F	LDAA	#1
2073		STAA	XMTRQ3
2074		STAA	GROUP3
2075		STAA	XFER3
2076		STAA	FLBFG3
2077		LDAA	SIGACQ
2078		ORAA	#040H
2079		ANDA	#0FBH
2080		STAA	SIGACQ
2081		STAA	LATCH1
2082		LDAA	MRFTA
2083		ORAA	#4
2084		STAA	LATCH3
2085		CLR	IO3+2
2086		ANDA	#0FBH
2087		ORAA	#040H
2088		STAA	MRFTA
2089		STAA	LATCH3
2090		LDAA	BPS16K
2091		BEQ	IO3FA
2092		LDX	#SBUFF1+6
2093		STX	INPT3
2094		LDX	#SBUFF1
2095		BRA	IO3FB
2096	IO3FA	LDX	#SBUFF3+6
2097		STX	INPT3
2098		LDX	#SBUFF3
2099	IO3FB	STX	OUTPT3
2100		LDAA	#0FFH
2101		STAA	0,X

TEST: I/O FIFO UNDERFLOW

BRANCH: NO UNDERFLOW

GET HERE W/MULTIPLE

TX REQ PER FRAME

SET "TX ACK"

CHECK TX INHIBIT

INIT FOR PORT #3

DATA TRANSMISSION

SET I/O INPUT MODE

CLR "SIG. ACQ. #3"

TOGGLE FIFO'S RESET

CLR PIA-B SIDE

SET XMT ACK I/O #3

STUFF 6 ELASTIC

BUFFER CHARACTERS (CC)

2102		STAA	1,X	
2103		STAA	2,X	
2104		STAA	3,X	
2105		STAA	4,X	
2106		STAA	5,X	
2107		LDX	#6	
2108		STX	CHCT3	
2109		JMP	IO4PT	
2110	IO3G	LDAA	BPS16K	
2111		BEQ	IO3GA	
2112		LDX	#SBUFF1	16KBPS
2113		BRA	IO3GB	
2114	IO3GA	LDX	#SBUFF3	
2115	IO3GB	STX	INPT3	
2116		STX	OUTPT3	
2117		LDX	#0	
2118		STX	CHCT3	
2119		LDAA	SIGACQ	CLR "SIGNAL ACQ. I/O #3"
2120		ANDA	#0FBH	
2121		STAA	SIGACQ	
2122		STAA	LATCH1	
2123		JMP	IO4PT	
2124	IO3H	LDAA	DATA5	CHECK TX INHIBIT
2125		ANDA	#1	
2126		BEQ	IO3HA	
2127		JMP	IO3YA1	
2128	IO3HA	LDAB	FLBFG3	
2129		BEQ	IO3HB	
2130		JMP	IO3O	
2131	IO3HB	LDAA	DATA6	TEST FIFO "O.R."
2132		BITA	#2	
2133		BNE	IO3I	
2134		JMP	IO3XP	
2135	IO3I	JSR	IO3LS	LOAD & STORE 1 CHAR
2136		LDAA	DATA6	
2137		BITA	#2	
2138		BNE	IO3K	
2139		JMP	IO3XP	
2140	IO3K	JSR	IO3LS	LOAD & STORE 2 CHAR
2141		LDAA	DATA6	
2142		BITA	#2	
2143		BNE	IO3M	
2144		JMP	IO3XP	
2145	IO3M	JSR	IO3LS	LOAD & STORE 3 CHAR
2146		LDAA	DATA6	
2147		BITA	#2	
2148		BNE	IO3N2	
2149		JMP	IO3XP	
2150	IO3N2	JSR	IO3LS	LOAD & STORE 4 CHAR
2151		LDAA	BPS16K	
2152		BNE	IO3N3	
2153		JMP	IO3XP	
2154	IO3N3	LDAA	DATA6	*16KBPS: ALLOW MORE*
2155		BITA	#2	*I/O SERVICING*
2156		BNE	IO3N4	
2157		JMP	IO3XP	
2158	IO3N4	JSR	IO3LS	LOAD & STORE 5 CHAR.

2159		LDAA	DATA6	
2160		BITA	#2	
2161		BNE	IO3N5	
2162		JMP	IO3XP	
2163	IO3N5	JSR	IO3LS	LOAD & STORE 6 CHAR.
2164		LDAA	DATA6	
2165		BITA	#2	
2166		BNE	IO3N6	
2167		JMP	IO3XP	
2168	IO3N6	JSR	IO3LS	LOAD & STORE 7 CHAR.
2169		LDAA	DATA6	
2170		BITA	#2	
2171		BNE	IO3N7	
2172		JMP	IO3XP	
2173	IO3N7	JSR	IO3LS	LOAD & STORE 8 CHAR.
2174		LDAA	DATA6	
2175		BITA	#2	
2176		BNE	IO3N8	
2177		JMP	IO3XP	
2178	IO3N8	JSR	IO3LS	LOAD & STORE 9 CHAR.
2179		LDAA	DATA6	
2180		BITA	#2	
2181		BNE	IO3N9	
2182		JMP	IO3XP	
2183	IO3N9	JSR	IO3LS	LOAD & STORE 10 CHAR.
2184		LDAA	DATA6	
2185		BITA	#2	
2186		BNE	IO3N10	
2187		JMP	IO3XP	
2188	IO3N10	JSR	IO3LS	LOAD & STORE 11 CHAR.
2189		JMP	IO3XP	
2190	IO3O	LDAA	DATA6	
2191		BITA	#2	
2192		BEQ	IO3XP	
2193	IO3P	CLR	FLBFG3	
2194		LDAA	PA1AD	SET "DATA RDY PORT #3"
2195		ORAA	#4	
2196		STAA	PA1AD	
2197		STAA	PIA1AD	
2198		JMP	IO3I	
2199	IO3XP	LDAA	EMPTY3	**TEST IF SETUP**
2200		BPL	IO3XQ	**LAST GOOD DATA PNTR**
2201		LDX	INPT3	DO SETUP
2202		STX	LGDP3	
2203		LDAA	#1	
2204		STAA	EMPTY3	
2205	IO3XQ	JMP	IO4PT	
2206	IO3Y	LDAA	XMTRQ3	
2207		BEQ	IO3YB	
2208		LDAA	MRFTA	
2209		ANDA	#0BFH	CLR "TX ACK"
2210		STAA	MRFTA	
2211		STAA	LATCH3	
2212		CLR	XMTRQ3	FIRST TIME THAT
2213		LDAA	#0FFH	TX REQ = FALSE
2214		STAA	EMPTY3	
2215		LDAA	PA1AD	

2216	ANDA	#4	
2217	BNE	IO3YA	
2218	STAA	CHCT3	NO DATA: ABORT INPUT
2219	STAA	CHCT3+1	
2220	STAA	EMPTY3	
2221	INCA		
2222	STAA	FLBFG3	
2223	LDAA	MRFTA	
2224	ORAA	#4	
2225	STAA	MRFTA	
2226	STAA	LATCH3	
2227	JMP	IO4PT	
2228	IO3YA	JMP	IO3D
2229	IO3YB	LDAA	EMPTY3
2230	BNE	IO3YA	
2231	JMP	IO3YD	
2232	IO3YA1	CLRA	
2233	STAA	XMTRQ3	
2234	STAA	EMPTY3	
2235	LDAA	PA1AD	
2236	ANDA	#0FBH	
2237	STAA	PIA1AD	CLEAR "DATA RDY PORT #3"
2238	STAA	PA1AD	
2239	LDAA	MRFTA	
2240	ORAA	#4	RESET I/O #1 FIFO
2241	ANDA	#0BFH	CLR TX ACK I/O #1
2242	STAA	MRFTA	
2243	STAA	LATCH3	
2244	IO3YC	LDAA	MRFTA
2245	ORAA	#4	**INIT I/O PORT**
2246	STAA	MRFTA	**TO OUTPUT**
2247	STAA	LATCH3	**RX BURST DATA**
2248	LDAA	SIGACQ	RESET I/O #3 FIFO
2249	ANDA	#0BFH	SELECT I/O OUTPUT MODE
2250	STAA	SIGACQ	
2251	STAA	LATCH1	
2252	IO3YC1	CLR	FMSDO3
2253	LDAA	#2	
2254	STAA	MSDO3	
2255	STAA	FLBFG3	
2256	JMP	IO3G	
2257	IO3YD	LDAA	FLBFG3
2258	BEQ	IO3YF	
2259	STAA	XFER3	
2260	LDAA	CHCT3+1	
2261	CMPA	#10	RCV ELASTIC BUFFER FULL?
2262	BGE	IO3YE	
2263	JMP	IO4PT	
2264	IO3YE	CLR	FLBFG3
2265	STAA	UT3	
2266	LDAA	SIGACQ	SET OUTPUT MODE
2267	ANDA	#0BFH	
2268	STAA	SIGACQ	
2269	STAA	LATCH1	
2270	LDAA	MRFTA	
2271	ANDA	#0FBH	CLR RESET ON I/O FIFO
2272	STAA	MRFTA	

2273	STAA	LATCH3	
2274	JSR	LD3	
2275	JSR	LD3	LOAD FIFO W/6 BUF. CH.
2276	LDAA	BPS16K	
2277	BEQ	IO3YE0	
2278	JSR	LD3	
2279	JSR	LD3	
2280	JSR	LD3	
2281	JSR	LD3	
2282	JSR	LD3	LOAD FIFO W/21 BUF. CH.
2283	IO3YE0 JMP	IO4PT	
2284	IO3YE1 LDAA	#1	
2285	STAA	XFER3	
2286	LDAA	UT3	
2287	BNE	IO3YE0	
2288	LDAA	DATA6	CHECK FIFO UNDERFLOW
2289	BITA	#2	BY TESTING FIFO "O.R."
2290	BEQ	IO3YC1	**UNDERFLOW**
2291	JMP	IO4PT	
2292	IO3YF LDX	CHCT3	
2293	BEQ	IO3YE1	
2294	LDAA	DATA6	
2295	LDAB	UT3	
2296	BNE	IO3YFA	
2297	BITA	#2	
2298	BEQ	IO3YC1	
2299	IO3YFA BITA	#020H	CHECK "I.R.": FULL?
2300	BEQ	IO4PT	
2301	IO3YG JSR	IO3LF	OUTPUT 1 CH. TO FIFO
2302	LDAA	DATA6	
2303	BITA	#020H	CHECK "I.R.": FULL?
2304	BEQ	IO4PT	
2305	IO3YH JSR	IO3LF	OUTPUT 2 CH. TO FIFO
2306	LDAA	DATA6	
2307	BITA	#020H	CHECK "I.R.": FULL?
2308	BEQ	IO4PT	
2309	IO3YI JSR	IO3LF	OUTPUT 3 CH. TO FIFO
2310	LDAA	BPS16K	
2311	BEQ	IO4PT	
2312	IO3YJ LDAA	DATA6	16 KBPS: SERVICE 10 CHARS.
2313	BITA	#020H	CHECK "I.R.": FULL?
2314	BEQ	IO4PT	
2315	IO3YK JSR	IO3LF	OUTPUT 4 CH. TO FIFO
2316	LDAA	DATA6	
2317	BITA	#020H	CHECK "I.R.": FULL?
2318	BEQ	IO4PT	
2319	IO3YL JSR	IO3LF	OUTPUT 5 CH. TO FIFO
2320	LDAA	DATA6	
2321	BITA	#020H	CHECK "I.R.": FULL?
2322	BEQ	IO4PT	
2323	IO3YM JSR	IO3LF	OUTPUT 6 CH. TO FIFO
2324	LDAA	DATA6	
2325	BITA	#020H	CHECK "I.R.": FULL?
2326	BEQ	IO4PT	
2327	IO3YN JSR	IO3LF	OUTPUT 7 CH. TO FIFO
2328	LDAA	DATA6	
2329	BITA	#020H	CHECK "I.R.": FULL?

```

2330      BEQ      IO4PT
2331 IO3YO JSR      IO3LF      OUTPUT 8 CH. TO FIFO
2332      LDAA     DATA6
2333      BITA     #020H      CHECK "I.R.": FULL?
2334      BEQ      IO4PT
2335 IO3YP JSR      IO3LF      OUTPUT 9 CH. TO FIFO
2336      LDAA     DATA6
2337      BITA     #020H      CHECK "I.R.": FULL?
2338      BEQ      IO4PT
2339 IO3YQ JSR      IO3LF      OUTPUT 10 CH. TO FIFO
2340 *****
2341 *
2342 *
2343 *
2344 IO4PT LDAA     PTONRX      PORT #4 ON?
2345      ANDA     #8
2346      BEQ      IO4A1
2347      CLR      P4OFF
2348      JMP      IO4A
2349 IO4A1 LDAA     P4OFF      PORT IS OFF
2350      BEQ      IO4A2
2351      JMP      BSERV
2352 IO4A2 STAA     CHCT4
2353      STAA     CHCT4+1
2354      STAA     XMTRQ4
2355      STAA     UT4
2356      LDX      #0
2357      STX      LGDPT4
2358      STAA     OUFG4
2359      STAA     EMPTY4
2360      LDAA     #2
2361      STAA     MSDO4
2362      INCA
2363      STAA     FLBFG4
2364      LDAA     MRFTA
2365      ORAA     #8          RESET FIFO I/O #4
2366      ANDA     #07FH      CLR XMT ACK
2367      STAA     MRFTA
2368      STAA     LATCH3
2369      STAA     P4OFF
2370      LDAA     PA1AD
2371      ANDA     #0F7H      CLEAR "DATA RDY PORT #4"
2372      STAA     PA1AD
2373      STAA     PIA1AD
2374      LDX      #SBUFF4
2375      STX      INPT4
2376      STX      OUTPT4
2377      LDAA     SIGACQ      CLR "SIGNAL ACQ. I/O #4"
2378      ANDA     #0F7H
2379      STAA     SIGACQ
2380      STAA     LATCH1
2381      LDX      TXPNTR
2382      CPX      #INPT4
2383      BNE      IO4A3
2384      CLR      SOEN
2385 IO4A3 JMP      BSERV
2386 *****

```



```

2387 *
2388 *   INPUT SERVICE SUBROUTINE
2389 *
2390 IO4LS  LDAA  #1
2391        STAA  XFER4
2392        LDAA  IO4      OBTAIN CHAR
2393        LDX   INPT4    GET PUT PNTR
2394        STAA  0,X      PUT IT
2395        INX
2396        CPX   OUTPT4   TEST OVERFLOW
2397        BNE   IO4LSX   BRANCH NO OVERFLOW
2398        LDAA  MRFTA    **OVERFLOW**
2399        ANDA  #07FH    CLR XMT ACK
2400        ORAA  #8       RESET FIFO
2401        STAA  MRFTA
2402        STAA  LATCH3
2403        STAA  OUF4     SET OUF4 FLAG
2404        INS
2405        INS          *REMOVE RETURN
2406        JMP   BSERV
2407 IO4LSX LDAA  BPS16K
2408        BEQ   IO4LSA
2409        CPX   #EBUFF4   16 KBPS
2410        BNE   IO4LSB
2411        LDX   #SBUFF1
2412        BRA   IO4LSB
2413 IO4LSA CPX   #EBUFF4   TEST END OF INPUT
2414        BNE   IO4LSB   NO
2415        LDX   #SBUFF4   GET INITIAL VALUE
2416 IO4LSB STX   INPT4
2417        LDX   CHCT4    GET CHAR COUNT
2418        INX
2419        STX   CHCT4    INCR COUNT
2420        RTS          PUT BACK
2421 *****
2422 *
2423 *LOAD I/O #4 FIFO WITH 1 CHAR. ROUTINE
2424 *
2425 IO4LF  LDX   OUTPT4
2426        LDAA  RXINHB
2427        BITA  #8        IS THE EXTEND TRANSMIT FLAG SET?
2428        BNE   IO4LF1    YES:RX NOT ALLOWED
2429        LDAA  0,X
2430        STAA  IO4+2     OUTPUT CH. TO I/O #4 FIFO
2431 IO4LF1 INX          ADJ. OUTPUT PNTR.
2432        LDAB  BPS16K
2433        BEQ   IO4LFA
2434        CPX   #EBUFF4   16KBPS
2435        BNE   IO4LFB
2436        LDX   #SBUFF1
2437        BRA   IO4LFB
2438 IO4LFA CPX   #EBUFF4
2439        BNE   IO4LFB
2440        LDX   #SBUFF4
2441 IO4LFB STX   OUTPT4
2442        LDAA  #1
2443        STAA  XFER4

```

```

2444      LDAA    FMSDO4    TEST MISSING DO
2445      BEQ     IO4LFC
2446      LDX     MDOCT4
2447      DEX
2448      STX     MDOCT4
2449      BNE     IO4LFC
2450      LDAA    SIGACQ
2451      ANDA    #0F7H
2452      STAA    SIGACQ
2453      STAA    LATCH1
2454      CLR     FMSDO4
2455 IO4LFC LDX     CHCT4    ADJUST CHR COUNT
2456      DEX
2457      STX     CHCT4
2458      BNE     IO4LFD
2459      INS
2460      INS
2461      JMP     IO4YE1    GO CHECK UNDERFLOW
2462 IO4LFD RTS
2463 *****
2464 *
2465 *LOAD FIFO I/O #4 WITH 3 BUF. CH. ROUTINE
2466 *
2467 LD4      LDX     OUTPT4
2468      LDAA    0,X
2469      STAA    IO4+2
2470      LDAA    1,X
2471      STAA    IO4+2
2472      LDAA    2,X
2473      STAA    IO4+2
2474      INX
2475      INX
2476      INX
2477      STX     OUTPT4
2478      LDX     CHCT4
2479      DEX
2480      DEX
2481      DEX
2482      STX     CHCT4
2483      RTS
2484 *****
2485 *
2486 ***** BEGIN SERVICING I/O PORT #4 *****
2487 *
2488 IO4A     LDAA    PTONRX    RX ONLY?
2489      BITA    #080H
2490      BEQ     IO4AA
2491      JMP     IO4Y
2492 IO4AA    LDAA    DATA1
2493      BITA    #8
2494      BNE     IO4B
2495      JMP     IO4Y
2496 IO4B     ASLA
2497      BPL     IO4C
2498      LDAA    XMTRQ4
2499      BNE     IO4C
2500      LDAA    EMPTY4

```

2501	BNE	IO4CA	
2502	LDX	CHCT4	R/T SELECTED: CHECK RCV DON
2503	BEQ	IO4BB	
2504	IO4BA	JMP	IO4YD
2505	IO4BB	LDAA	DATA6 TEST: I/O FIFO UNDERFLOW
2506		BITA	#1
2507	BNE	IO4BA	BRANCH: NO UNDERFLOW
2508	IO4C	LDAA	XMTRQ4
2509		BNE	IO4D
2510		LDAA	EMPTY4
2511		BEQ	IO4E
2512	IO4CA	CLR	EMPTY4 GET HERE W/MULTIPLE
2513		STAA	XMTRQ4 TX REQ PER FRAME
2514		LDX	#0
2515		STX	LGDP4
2516		LDAA	MRFTA
2517		ORAA	#080H SET "TX ACK"
2518		STAA	MRFTA
2519		STAA	LATCH3
2520	IO4D	LDAA	OUG4
2521		BNE	IO4DA
2522		JMP	IO4H
2523	IO4DA	STAA	XFER4
2524		JMP	BSERV
2525	IO4E	LDAA	DATA5 CHECK TX INHIBIT
2526		ANDA	#1
2527		BEQ	IO4F
2528		JMP	BSERV
2529	IO4F	LDAA	#1 **INIT FOR PORT #4**
2530		STAA	XMTRQ4 **DATA TRANSMISSION**
2531		STAA	GROUP4
2532		STAA	XFER4
2533		STAA	FLBFG4
2534		LDAA	SIGACQ
2535		ORAA	#080H SELECT I/O INPUT MODE
2536		ANDA	#0F7H CLR "SIG. ACQ. #4"
2537		STAA	SIGACQ
2538		STAA	LATCH1
2539		LDAA	MRFTA
2540		ORAA	#8 TOGGLE FIFO'S RESET
2541		STAA	LATCH3
2542		CLR	IO4+2 CLR PIA-B SIDE
2543		ANDA	#0F7H
2544		ORAA	#080H SET XMT ACK I/O #4
2545		STAA	MRFTA
2546		STAA	LATCH3
2547		LDAA	BPS16K
2548		BEQ	IO4FA
2549		LDX	#SBUFF1+6 16KBPS
2550		STX	INPT4
2551		LDX	#SBUFF1
2552		BRA	IO4FB
2553	IO4FA	LDX	#SBUFF4+6
2554		STX	INPT4
2555		LDX	#SBUFF4
2556	IO4FB	STX	OUTPT4
2557		LDAA	#0FFH STUFF 6 ELASTIC

2558	STAA	0,X	BUFFER CHARACTERS (CC)
2559	STAA	1,X	
2560	STAA	2,X	
2561	STAA	3,X	
2562	STAA	4,X	
2563	STAA	5,X	
2564	LDX	#6	
2565	STX	CHCT4	
2566	JMP	BSERV	
2567	IO4G	LDAA	BPS16K
2568		BEQ	IO4GA
2569		LDX	#SBUFF1 16 KBPS
2570		BRA	IO4GB
2571	IO4GA	LDX	#SBUFF4
2572	IO4GB	STX	INPT4
2573		STX	OUTPT4
2574		LDX	#0
2575		STX	CHCT4
2576		LDAA	SIGACQ CLR "SIGNAL ACQ. I/O #4"
2577		ANDA	#0F7H
2578		STAA	SIGACQ
2579		STAA	LATCH1
2580		JMP	BSERV
2581	IO4H	LDAA	DATA5 CHECK TX INHIBIT
2582		ANDA	#1
2583		BEQ	IO4HA
2584		JMP	IO4YA1
2585	IO4HA	LDAB	FLBFG4
2586		BEQ	IO4HB
2587		JMP	IO4O
2588	IO4HB	LDAA	DATA6 TEST FIFO "O.R."
2589		BITA	#1
2590		BNE	IO4I
2591		JMP	IO4XP
2592	IO4I	JSR	IO4LS LOAD & STORE 1 CHAR
2593		LDAA	DATA6
2594		BITA	#1
2595		BNE	IO4K
2596		JMP	IO4XP
2597	IO4K	JSR	IO4LS LOAD & STORE 2 CHAR
2598		LDAA	DATA6
2599		BITA	#1
2600		BNE	IO4M
2601		JMP	IO4XP
2602	IO4M	JSR	IO4LS LOAD & STORE 3 CHAR.
2603		LDAA	DATA6
2604		BITA	#1
2605		BNE	IO4N2
2606		JMP	IO4XP
2607	IO4N2	JSR	IO4LS LOAD & STORE 4 CHAR
2608		LDAA	BPS16K
2609		BNE	IO4N3
2610		JMP	IO4XP
2611	IO4N3	LDAA	DATA6 *16 KBPS: ALLOW MORE*
2612		BITA	#1 *I/O SERVICING*
2613		BNE	IO4N4
2614		JMP	IO4XP

2615	IO4N4	JSR	IO4LS	LOAD & STORE 5 CHAR.
2616		LDAA	DATA6	
2617		BITA	#1	
2618		BNE	IO4N5	
2619		JMP	IO4XP	
2620	IO4N5	JSR	IO4LS	LOAD & STORE 6 CHAR.
2621		LDAA	DATA6	
2622		BITA	#1	
2623		BNE	IO4N6	
2624		JMP	IO4XP	
2625	IO4N6	JSR	IO4LS	LOAD & STORE 7 CHAR.
2626		LDAA	DATA6	
2627		BITA	#1	
2628		BNE	IO4N7	
2629		JMP	IO4XP	
2630	IO4N7	JSR	IO4LS	LOAD & STORE 8 CHAR.
2631		LDAA	DATA6	
2632		BITA	#1	
2633		BNE	IO4N8	
2634		JMP	IO4XP	
2635	IO4N8	JSR	IO4LS	LOAD & STORE 9 CHAR.
2636		LDAA	DATA6	
2637		BITA	#1	
2638		BNE	IO4N9	
2639		JMP	IO4XP	
2640	IO4N9	JSR	IO4LS	LOAD & STORE 10 CHAR.
2641		LDAA	DATA6	
2642		BITA	#1	
2643		BNE	IO4N10	
2644		JMP	IO4XP	
2645	IO4N10	JSR	IO4LS	LOAD & STORE 11 CHAR.
2646		JMP	IO4XP	
2647	IO4O	LDAA	DATA6	
2648		BITA	#1	
2649		BEQ	IO4XP	
2650	IO4P	CLR	FLBFG4	
2651		LDAA	PA1AD	SET "DATA RDY PORT #4"
2652		ORAA	#8	
2653		STAA	PA1AD	
2654		STAA	PIA1AD	
2655		JMP	IO4I	
2656	IO4XP	LDAA	EMPTY4	**TEST IF SETUP**
2657		BPL	IO4XQ	**LAST GOOD DATA PNTR**
2658		LDX	INPT4	DO SETUP
2659		STX	LGDP4	
2660		LDAA	#1	
2661		STAA	EMPTY4	
2662	IO4XQ	JMP	BSERV	
2663	IO4Y	LDAA	XMTRQ4	
2664		BEQ	IO4YB	
2665		LDAA	MRFTA	
2666		ANDA	#07FH	CLR "TX ACK"
2667		STAA	MRFTA	
2668		STAA	LATCH3	
2669		CLR	XMTRQ4	FIRST TIME THAT
2670		LDAA	#0FFH	TX REQ = FALSE
2671		STAA	EMPTY4	

2672	LDAA	PA1AD	
2673	ANDA	#8	
2674	BNE	IO4YA	
2675	STAA	CHCT4	NO DATA: ABORT INPUT
2676	STAA	CHCT4+1	
2677	STAA	EMPTY4	
2678	INCA		
2679	STAA	FLBFG4	
2680	LDAA	MRFTA	
2681	ORAA	#8	
2682	STAA	MRFTA	
2683	STAA	LATCH3	
2684	JMP	BSERV	
2685	IO4YA	JMP	IO4D
2686	IO4YB	LDAA	EMPTY4
2687	BNE	IO4YA	
2688	JMP	IO4YD	
2689	IO4YA1	CLRA	
2690	STAA	XMTRQ4	
2691	STAA	EMPTY4	
2692	LDAA	PA1AD	
2693	ANDA	#0F7H	
2694	STAA	PIA1AD	CLEAR "DATA RDY PORT #4"
2695	STAA	PA1AD	
2696	LDAA	MRFTA	
2697	ORAA	#8	RESET I/O #4 FIFO
2698	ANDA	#07FH	CLR TX ACK I/O #4
2699	STAA	MRFTA	
2700	STAA	LATCH3	
2701	IO4YC	LDAA	MRFTA
2702	ORAA	#8	**INIT I/O PORT**
2703	STAA	MRFTA	**TO OUTPUT**
2704	STAA	LATCH3	**RX BURST DATA**
2705	LDAA	SIGACQ	RESET I/O #4 FIFO
2706	ANDA	#07FH	
2707	STAA	SIGACQ	SELECT I/O OUTPUT MODE
2708	STAA	LATCH1	
2709	IO4YC1	CLR	FMSDO4
2710	LDAA	#2	
2711	STAA	MSDO4	
2712	STAA	FLBFG4	
2713	JMP	IO4G	
2714	IO4YD	LDAA	FLBFG4
2715	BEQ	IO4YF	
2716	STAA	XFER4	
2717	LDAA	CHCT4+1	
2718	CMPA	#10	RCV ELASTIC BUFFER FULL?
2719	BGE	IO4YE	
2720	JMP	BSERV	
2721	IO4YE	CLR	FLBFG4
2722	STAA	UT4	
2723	LDAA	SIGACQ	SET OUTPUT MODE
2724	ANDA	#07FH	
2725	STAA	SIGACQ	
2726	STAA	LATCH1	
2727	LDAA	MRFTA	
2728	ANDA	#0F7H	RESET ON I/O FIFO

2729	STAA	MRFTA	
2730	STAA	LATCH3	
2731	JSR	LD4	
2732	JSR	LD4	LOAD FIFO W/6 BUF. CH.
2733	LDAA	BPS16K	
2734	BEQ	IO4YE0	
2735	JSR	LD4	16 KBPS SO:
2736	JSR	LD4	
2737	JSR	LD4	
2738	JSR	LD4	
2739	JSR	LD4	LOAD FIFO W/21 BUF. CH.
2740	IO4YE0	JMP	BSERV
2741	IO4YE1	LDAA	#1
2742	STAA	XFER4	
2743	LDAA	UT4	
2744	BNE	IO4YE0	
2745	LDAA	DATA6	CHECK FIFO UNDERFLOW
2746	BITA	#1	BY TESTING FIFO "O.R."
2747	BEQ	IO4YC1	**UNDERFLOW**
2748	JMP	BSERV	
2749	IO4YF	LDX	CHCT4
2750	BEQ	IO4YE1	
2751	LDAA	DATA6	
2752	LDAB	UT4	
2753	BNE	IO4YFA	
2754	BITA	#1	
2755	BEQ	IO4YC1	
2756	IO4YFA	BITA	#010H CHECK "I.R.": FULL?
2757	BEQ	BSERV	
2758	IO4YG	JSR	IO4LF OUTPUT 1 CHAN. TO FIFO
2759	LDAA	DATA6	
2760	BITA	#010H	CHECK "I.R.": FULL?
2761	BEQ	BSERV	
2762	IO4YH	JSR	IO4LF OUTPUT 2 CH. TO FIFO
2763	LDAA	DATA6	
2764	BITA	#010H	CHECK "I.R.": FULL?
2765	BEQ	BSERV	
2766	IO4YI	JSR	IO4LF OUTPUT 3 CH. TO FIFO
2767	LDAA	BPS16K	
2768	BEQ	BSERV	
2769	IO4YJ	LDAA	DATA6 16KBPS: SERVICE 10 CHAR.
2770	BITA	#010H	CHECK "I.R.": FULL?
2771	BEQ	BSERV	
2772	IO4YK	JSR	IO4LF OUTPUT 4 CH. TO FIFO
2773	LDAA	DATA6	
2774	BITA	#010H	CHECK "I.R.": FULL?
2775	BEQ	BSERV	
2776	IO4YL	JSR	IO4LF OUTPUT 5 CH. TO FIFO
2777	LDAA	DATA6	
2778	BITA	#010H	CHECK "I.R.": FULL?
2779	BEQ	BSERV	
2780	IO4YM	JSR	IO4LF OUTPUT 6 CH. TO FIFO
2781	LDAA	DATA6	
2782	BITA	#010H	CHECK "I.R.": FULL?
2783	BEQ	BSERV	
2784	IO4YN	JSR	IO4LF OUTPUT 7 CH. TO FIFO
2785	LDAA	DATA6	

```

2786      BITA    #010H    CHECK "I.R.": FULL?
2787      BEQ     BSERV
2788 IO4YO  JSR     IO4LF    OUTPUT 8 CH. TO FIFO
2789      LDAA    DATA6
2790      BITA    #010H    CHECK "I.R.": FULL?
2791      BEQ     BSERV
2792 IO4YP  JSR     IO4LF    OUTPUT 9 CH. TO FIFO
2793      LDAA    DATA6
2794      BITA    #010H    CHECK "I.R.": FULL?
2795 IO4YQ  JSR     IO4LF    OUTPUT 10 CH. TO FIFO
2796 *****
2797 *
2798 *   BITE SERVICE ROUTINE
2799 *
2800 *
2801 BSERV  LDAA    DATA5
2802      BITA    #8        TEST:  SYSTEM BITE RESET
2803      BEQ     BSERVA
2804      CLR     IRSBIT
2805      CLR     BITELT
2806 BSERVA BITA    #04H    TEST:  LAMP TEST
2807      BEQ     BSERVB
2808      LDAA    #0FFH    TURN ALL BITE LITES ON
2809      BRA     BSERVC
2810 *
2811 *
2812 *
2813 *
2814 *
2815 *
2816 *
2817 *
2818 *
2819 *
2820 *
2821 BSERVB LDAA    RXINHB  NO ERROR FOR PORT WITH
2822      ASLA
2823      COMA
2824      ANDA    BITELT    OUTPUT ALL OTHER ERRORS PRESENT
2825      STAA    BITELT
2826 BSERVC STAA    LATCH2
2827      LDAB    DATA5
2828      LDAA    TIMEFG
2829      BEQ     BSERVF
2830      BITB    #010H    TEST 75/16 HZ CLOCK
2831      BNE     BSERVE
2832 BSERVD CLR     TIMEFG
2833 BSERVE JMP     SOF
2834 BSERVF BITB    #010H    TEST 75/16 HZ CLOCK
2835      BEQ     BSERVE
2836      INC     TIMEFG
2837      LDAB    BITELT
2838      LDX     CHCT1
2839      BEQ     SERVF3
2840 SERVF1 LDAA    XFER1
2841      BEQ     SERVF2
2842      CLR     XFER1

```



```

2843      BRA      BSERVG
2844 SERV2 ORAB    #2      I/O #1 BITE ERROR
2845      BRA      BSERVG
2846 SERV3 LDAA    XMTRQ1   NO DATA AFTER TX REQ?
2847      BNE      SERV1
2848 BSERVG LDX     CHCT2
2849      BEQ      SERV3
2850 SERV1 LDAA    XFER2
2851      BEQ      SERV2
2852      CLR      XFER2
2853      BRA      BSERVH
2854 SERV2 ORAB    #4      I/O #2 BITE ERROR
2855      BRA      BSERVH
2856 SERV3 LDAA    XMTRQ2   NO DATA AFTER TX REQ?
2857      BNE      SERV1
2858 BSERVH LDX     CHCT3
2859      BEQ      SERV3
2860 SERV1 LDAA    XFER3
2861      BEQ      SERV2
2862      CLR      XFER3
2863      BRA      BSERVI
2864 SERV2 ORAB    #8      I/O #3 BITE ERROR
2865      BRA      BSERVI
2866 SERV3 LDAA    XMTRQ3   NO DATA AFTER TX REQ?
2867      BNE      SERV1
2868 BSERVI LDX     CHCT4
2869      BEQ      SERV3
2870 SERV1 LDAA    XFER4
2871      BEQ      SERV2
2872      CLR      XFER4
2873      BRA      BSERVK
2874 SERV2 ORAB    #010H    I/O #4 BITE ERROR
2875      BRA      BSERVK
2876 SERV3 LDAA    XMTRQ4   NO DATA AFTER TX REQ?
2877      BNE      SERV1
2878 BSERVK STAB    BITELT
2879 *****
2880 *
2881 *      BEGIN SERVICING SYSTEM OUTPUT FIFO
2882 *
2883 SOF     LDAA    SOEN
2884      BEQ      SOF1
2885      BSR      SOFA
2886 SOF1    JMP     SIF
2887 *
2888 SOF2    LDX     OUTCHC   *TX SELF TEST ROUTINE*
2889      LDAB     #16
2890 SOF3    LDAA    DATA5   FIFO RDY?
2891      BMI      SOF5
2892      LDAA     #033H      OUTPUT TEST CHAR.
2893      STAA     PIA1BD
2894      DEX
2895      BNE      SOF4
2896      CLRA
2897      STAA     SOEN
2898      STAA     SELFTT
2899      STX      OUTCHC

```

2900		RTS		
2901	SOF4	DECB		
2902		BNE	SOF3	
2903	SOF5	STX	OUTCHC	
2904		RTS		
2905	*			
2906	SOFA	LDAA	SELFTT	TX SELF TEST?
2907		BNE	SOF2	
2908		LDX	TXPNTR	
2909		LDAA	DATA5	CHECK TX INHIBIT
2910		BITA	#1	
2911		BEQ	SOFAA	
2912		CLR	SOEN	TX INHIBIT: ABORT
2913		RTS		
2914	SOFAA	LDX	2,X	
2915		STX	TBPNTR	
2916		LDX	TXPNTR	
2917		LDX	4,X	
2918		STX	TXCHCT	
2919		LDX	TXPNTR	
2920		LDX	14,X	SETUP L.G.D. PNTR FOR
2921		STX	TXLGDP	TX BURST SERVICE
2922		LDAA	GRP1FG	
2923		BNE	SOFB	
2924		JMP	SOFY	
2925	SOFB	LDX	FILLCT	
2926		LDAB	#OFFH	FILL PATTERN BEFORE DATA
2927		LDAA	DATA5	FIFO RDY? (DATA5-H=LOW)
2928		BPL	SOFBA	
2929		STX	FILLCT	
2930		RTS		
2931	SOFBA	STAB	PIA1BD	OUTPUT FILL CHAR #1
2932		DEX		
2933		BNE	SOFBB	
2934		CLR	GRP1FG	
2935		JMP	SOFD+2	
2936	SOFBB	LDAA	DATA5	FIFO READY?
2937		BPL	SOFBC	
2938		STX	FILLCT	
2939		RTS		
2940	SOFBC	STAB	PIA1BD	OUTPUT FILL CHAR #2
2941		DEX		
2942		BNE	SOFBD	
2943		CLR	GRP1FG	
2944		JMP	SOFD+2	
2945	SOFBD	LDAA	DATA5	FIFO READY?
2946		BPL	SOFBE	
2947		STX	FILLCT	
2948		RTS		
2949	SOFBE	STAB	PIA1BD	OUTPUT FILL CHAR #3
2950		DEX		
2951		BNE	SOFBF	
2952		CLR	GRP1FG	
2953		JMP	SOFD+2	
2954	SOFBF	LDAA	DATA5	FIFO READY?
2955		BPL	SOFBG	
2956		STX	FILLCT	

2957		RTS	
2958	SOFBG	STAB	PIA1BD OUTPUT FILL CHAR #4
2959		DEX	
2960		BNE	SOFBH
2961		CLR	GRP1FG
2962		JMP	SFG+2
2963	SOFBH	LDAA	DATA5 FIFO READY?
2964		BPL	SOFBI
2965		STX	FILLCT
2966		RTS	
2967	SOFBI	STAB	PIA1BD OUTPUT FILL CHAR #5
2968		DEX	
2969		BNE	SOFBJ
2970		CLR	GRP1FG
2971		JMP	SFH+2
2972	SOFBJ	LDAA	DATA5 FIFO READY?
2973		BPL	SOFBK
2974		STX	FILLCT
2975		RTS	
2976	SOFBK	STAB	PIA1BD OUTPUT FILL CHAR #6
2977		DEX	
2978		BNE	SOFBL
2979		CLR	GRP1FG
2980		JMP	SFI+2
2981	SOFBL	LDAA	DATA5 FIFO READY?
2982		BPL	SOFBM
2983		STX	FILLCT
2984		RTS	
2985	SOFBM	STAB	PIA1BD OUTPUT FILL CHAR #7
2986		DEX	
2987		BNE	SOFBN
2988		CLR	GRP1FG
2989		JMP	SFJ+2
2990	SOFBN	LDAA	DATA5 FIFO READY?
2991		BPL	SOFBO
2992		STX	FILLCT
2993		RTS	
2994	SOFBO	STAB	PIA1BD OUTPUT FILL CHAR #8
2995		DEX	
2996		BNE	SOFBP
2997		CLR	GRP1FG
2998		JMP	SFK+2
2999	SOFBP	LDAA	DATA5 FIFO READY?
3000		BPL	SOFBQ
3001		STX	FILLCT
3002		RTS	
3003	SOFBQ	STAB	PIA1BD OUTPUT FILL CHAR #9
3004		DEX	
3005		BNE	SOFBR
3006		CLR	GRP1FG
3007		JMP	SFL+2
3008	SOFBR	LDAA	DATA5 FIFO READY?
3009		BPL	SOFBS
3010		STX	FILLCT
3011		RTS	
3012	SOFBS	STAB	PIA1BD OUTPUT FILL CHAR #10
3013		DEX	

3014	BNE	SOFBT	
3015	CLR	GRP1FG	
3016	JMP	SOFM+2	
3017	SOFBT	LDAA	DATA5 FIFO READY?
3018	BPL	SOFBU	
3019	STX	FILLCT	
3020	RTS		
3021	SOFBU	STAB	PIA1BD OUTPUT FILL CHAR #11
3022	DEX		
3023	BNE	SOFBV	
3024	CLR	GRP1FG	
3025	JMP	SOFN+2	
3026	SOFBV	LDAA	DATA5 FIFO READY?
3027	BPL	SOFBW	
3028	STX	FILLCT	
3029	RTS		
3030	SOFBW	STAB	PIA1BD OUTPUT FILL CHAR #12
3031	DEX		
3032	BNE	SOFBX	
3033	CLR	GRP1FG	
3034	JMP	SOFO+2	
3035	SOFBX	LDAA	DATA5 FIFO READY?
3036	BPL	SOFBY	
3037	STX	FILLCT	
3038	RTS		
3039	SOFBY	STAB	PIA1BD OUTPUT FILL CHAR #13
3040	DEX		
3041	BNE	SOFBZ	
3042	CLR	GRP1FG	
3043	JMP	SOFQ+2	
3044	SOFBZ	LDAA	DATA5 FIFO READY?
3045	BPL	SOFB1	
3046	STX	FILLCT	
3047	RTS		
3048	SOFB1	STAB	PIA1BD OUTPUT FILL CHAR #14
3049	DEX		
3050	BNE	SOFB2	
3051	CLR	GRP1FG	
3052	JMP	SOFQ+2	
3053	SOFB2	LDAA	DATA5 FIFO READY?
3054	BPL	SOFB3	
3055	STX	FILLCT	
3056	RTS		
3057	SOFB3	STAB	PIA1BD OUTPUT FILL CHAR #15
3058	DEX		
3059	BNE	SOFB4	
3060	CLR	GRP1FG	
3061	JMP	SOFQ+2	
3062	SOFB4	LDAA	DATA5 FIFO READY?
3063	BPL	SOFB5	
3064	STX	FILLCT	
3065	RTS		
3066	SOFB5	STAB	PIA1BD OUTPUT FILL CHAR #16
3067	DEX		
3068	BNE	SOFB6	
3069	CLR	GRP1FG	
3070	SOFB6	STX	FILLCT

3071		RTS		
3072	SOFC	LDAA	DATA5	FIFO READY?
3073		BPL	SOFCA	
3074		JMP	SOFS	
3075	SOFCA	LDX	TXCHCT	
3076		BNE	SOFCE	
3077		LDAA	#0FFH	GET STUFF CHAR
3078		STAA	TXEND	
3079		LDX	TXPNTR	
3080		STAA	13,X	SET PORT'S OUF
3081		BRA	SOFCD	
3082	SOFCB	DEX		
3083		STX	TXCHCT	
3084		LDX	TBPNTR	
3085		LDAA	0,X	GET BUFF CHAR
3086		INX		
3087		CPX	TXUVAL	ADJ BUFF PNTRS
3088		BNE	SOFCC	
3089		LDX	TXLVAL	
3090	SOFC	STX	TBPNTR	
3091		CPX	TXLGDP	LAST GOOD DATA?
3092		BNE	SOFCD	
3093		LDX	#0	YES
3094		STX	TXLGDP	
3095		STAB	TXDONE	
3096	SOFC	STAA	PIA1BD	OUTPUT CHAR #1
3097		LDX	OUTCHC	
3098		DEX		
3099		BNE	SOFD	
3100		JMP	SOFT	
3101	SOFD	STX	OUTCHC	
3102		LDAA	DATA5	FIFO READY?
3103		BPL	SOFDA	
3104		JMP	SOFS	
3105	SOFDA	LDX	TXCHCT	
3106		BNE	SOFDB	
3107		LDAA	#0FFH	GET STUFF CHAR
3108		STAA	TXEND	
3109		LDX	TXPNTR	
3110		STAA	13,X	
3111		BRA	SOFDD	
3112	SOFDB	DEX		
3113		STX	TXCHCT	
3114		LDX	TBPNTR	
3115		LDAA	0,X	GET BUFF CHAR
3116		INX		
3117		CPX	TXUVAL	ADJ BUFF PNTRS
3118		BNE	SOFDC	
3119		LDX	TXLVAL	
3120	SOFDC	STX	TBPNTR	
3121		CPX	TXLGDP	LAST GOOD DATA?
3122		BNE	SOFDD	
3123		LDX	#0	YES
3124		STX	TXLGDP	
3125		STAB	TXDONE	
3126	SOFDD	STAA	PIA1BD	OUTPUT CHAR #2
3127		LDX	OUTCHC	

3128	DEX		
3129	BNE	SOFE	
3130	JMP	SOFT	
3131	SOFE	STX	OUTCHC
3132	LDAA	DATA5	FIFO READY?
3133	BPL	SOFEA	
3134	JMP	SOFS	
3135	SOFEA	LDX	TXCHCT
3136	BNE	SOFEB	
3137	LDAA	#0FFH	GET STUFF CHAR
3138	STAA	TXEND	
3139	LDX	TXPNTR	
3140	STAA	13,X	SET PORT'S UOFG
3141	BRA	SOFED	
3142	SOFEB	DEX	
3143	STX	TXCHCT	
3144	LDX	TBPNTR	
3145	LDAA	0,X	GET BUFF CHAR
3146	INX		
3147	CPX	TXUVAL	ADJ BUFF PNTRS
3148	BNE	SOFEC	
3149	LDX	TXLVAL	
3150	SOFEC	STX	TBPNTR
3151	CPX	TXLGDP	LAST GOOD DATA?
3152	BNE	SOFED	
3153	LDX	#0	YES
3154	STX	TXLGDP	
3155	STAB	TXDONE	
3156	SOFED	STAA	PIA1BD
3157	LDX	OUTCHC	OUTPUT CHAR #3
3158	DEX		
3159	BNE	SOFF	
3160	JMP	SOFT	
3161	SOFF	STX	OUTCHC
3162	LDAA	DATA5	FIFO READY?
3163	BPL	SOFFA	
3164	JMP	SOFS	
3165	SOFFA	LDX	TXCHCT
3166	BNE	SOFFB	
3167	LDAA	#0FFH	GET STUFF CHAR
3168	STAA	TXEND	
3169	LDX	TXPNTR	
3170	STAA	13,X	SET PORT'S OUFG
3171	BRA	SOFFD	
3172	SOFFB	DEX	
3173	STX	TXCHCT	
3174	LDX	TBPNTR	
3175	LDAA	0,X	GET BUFF CHAR
3176	INX		
3177	CPX	TXUVAL	ADJ BUFF PNTRS
3178	BNE	SOFFC	
3179	LDX	TXLVAL	
3180	SOFFC	STX	TBPNTR
3181	CPX	TXLGDP	LAST GOOD DATA?
3182	BNE	SOFFD	
3183	LDX	#0	YES
3184	STX	TXLGDP	

3185	STAB	TXDONE	
3186	SOFFD	STAA	PIA1BD OUTPUT CHAR #4
3187	LDX	OUTCHC	
3188	DEX		
3189	BNE	SOFG	
3190	JMP	SOFT	
3191	SOFG	STX	OUTCHC
3192	LDAA	DATA5	FIFO READY?
3193	BPL	SOFGA	
3194	JMP	SOFS	
3195	SOFGA	LDX	TXCHCT
3196	BNE	SOFGB	
3197	LDAA	#0FFH	GET STUFF CHAR
3198	STAA	TXEND	
3199	LDX	TXPNTR	
3200	STAA	13,X	SET PORT'S OUFG
3201	BRA	SOFGD	
3202	SOFGB	DEX	
3203	STX	TXCHCT	
3204	LDX	TBPNTR	
3205	LDAA	0,X	GET BUFF CHAR
3206	INX		
3207	CPX	TXUVAL	ADJ BUFF PNTRS
3208	BNE	SOFGC	
3209	LDX	TXLVAL	
3210	SOFGC	STX	TBPNTR
3211	CPX	TXLGDP	LAST GOOD DATA?
3212	BNE	SOFGD	
3213	LDX	#0	YES
3214	STX	TXLGDP	
3215	STAB	TXDONE	
3216	SOFGD	STAA	PIA1BD OUTPUT CHAR #5
3217	LDX	OUTCHC	
3218	DEX		
3219	BNE	SOFH	
3220	JMP	SOFT	
3221	SOFH	STX	OUTCHC
3222	LDAA	DATA5	FIFO READY?
3223	BPL	SOFHA	
3224	JMP	SOFS	
3225	SOFHA	LDX	TXCHCT
3226	BNE	SOFHB	
3227	LDAA	#0FFH	GET STUFF CHAR
3228	STAA	TXEND	
3229	LDX	TXPNTR	
3230	STAA	13,X	SET PORT'S OUFG
3231	BRA	SOFHD	
3232	SOFHB	DEX	
3233	STX	TXCHCT	
3234	LDX	TBPNTR	
3235	LDAA	0,X	GET BUFF CHAR
3236	INX		
3237	CPX	TXUVAL	ADJ BUFF PNTRS
3238	BNE	SOFHC	
3239	LDX	TXLVAL	
3240	SOFHC	STX	TBPNTR
3241	CPX	TXLGDP	LAST GOOD DATA?

3242	BNE	SOFHD	
3243	LDX	#0	YES
3244	STX	TXLGDP	
3245	STAB	TXDONE	
3246	SOFHD STAA	PIA1BD	OUTPUT CHAR #6
3247	LDX	OUTCHC	
3248	DEX		
3249	BNE	SOFI	
3250	JMP	SOFT	
3251	SOFI STX	OUTCHC	
3252	LDAA	DATA5	FIFO READY?
3253	BPL	SOFIA	
3254	JMP	SOFS	
3255	SOFIA LDX	TXCHCT	
3256	BNE	SOFIB	
3257	LDAA	#0FFH	GET STUFF CHAR
3258	STAA	TXEND	
3259	LDX	TXPNTR	
3260	STAA	13,X	SET PORT'S OUFG
3261	BRA	SOFID	
3262	SOFIB DEX		
3263	STX	TXCHCT	
3264	LDX	TBPNTR	
3265	LDAA	0,X	GET BUFF CHAR
3266	INX		
3267	CPX	TXUVAL	ADJ BUFF PNTRS
3268	BNE	SOFIC	
3269	LDX	TXLVAL	
3270	SOFIC STX	TBPNTR	
3271	CPX	TXLGDP	LAST GOOD DATA?
3272	BNE	SOFID	
3273	LDX	#0	YES
3274	STX	TXLGDP	
3275	STAB	TXDONE	
3276	SOFID STAA	PIA1BD	OUTPUT CHAR #7
3277	LDX	OUTCHC	
3278	DEX		
3279	BNE	SOFJ	
3280	JMP	SOFT	
3281	SOFJ STX	OUTCHC	
3282	LDAA	DATA5	FIFO READY?
3283	BPL	SOFJA	
3284	JMP	SOFS	
3285	SOFJA LDX	TXCHCT	
3286	BNE	SOFJB	
3287	LDAA	#0FFH	GET STUFF CHAR
3288	STAA	TXEND	
3289	LDX	TXPNTR	
3290	STAA	13,X	SET PORT'S OUFG
3291	BRA	SOFJD	
3292	SOFJB DEX		
3293	STX	TXCHCT	
3294	LDX	TBPNTR	
3295	LDAA	0,X	GET BUFF CHAR
3296	INX		
3297	CPX	TXUVAL	ADJ BUFF PNTRS
3298	BNE	SOFJC	

3299		LDX	TXLVAL	
3300	SOFJC	STX	TBPNTR	
3301		CPX	TXLGDP	LAST GOOD DATA?
3302		BNE	SOFJD	
3303		LDX	#0	YES
3304		STX	TXLGDP	
3305		STAB	TXDONE	
3306	SOFJD	STAA	PIA1BD	OUTPUT CHAR #8
3307		LDX	OUTCHC	
3308		DEX		
3309		BNE	SOFK	
3310		JMP	SOFT	
3311	SOFK	STX	OUTCHC	
3312		LDAA	DATA5	FIFO READY?
3313		BPL	SOFKA	
3314		JMP	SOFS	
3315	SOFKA	LDX	TXCHCT	
3316		BNE	SOFKB	
3317		LDAA	#0FFH	GET STUFF CHAR
3318		STAA	TXEND	
3319		LDX	TXPNTR	
3320		STAA	13,X	SET PORT'S OUFG
3321		BRA	SOFKD	
3322	SOFKB	DEX		
3323		STX	TXCHCT	
3324		LDX	TBPNTR	
3325		LDAA	0,X	GET BUFF CHAR
3326		INX		
3327		CPX	TXUVAL	ADJ BUFF PNTRS
3328		BNE	SOFKC	
3329		LDX	TXLVAL	
3330	SOFKC	STX	TBPNTR	
3331		CPX	TXLGDP	LAST GOOD DATA?
3332		BNE	SOFKD	
3333		LDX	#0	YES
3334		STX	TXLGDP	
3335		STAB	TXDONE	
3336	SOFKD	STAA	PIA1BD	OUTPUT CHAR #9
3337		LDX	OUTCHC	
3338		DEX		
3339		BNE	SOFL	
3340		JMP	SOFT	
3341	SOFL	STX	OUTCHC	
3342		LDAA	DATA5	FIFO READY?
3343		BPL	SOFLA	
3344		JMP	SOFS	
3345	SOFLA	LDX	TXCHCT	
3346		BNE	SOFLB	
3347		LDAA	#0FFH	GET STUFF CHAR
3348		STAA	TXEND	
3349		LDX	TXPNTR	
3350		STAA	13,X	SET PORT'S OUFG
3351		BRA	SOFLD	
3352	SOFLB	DEX		
3353		STX	TXCHCT	
3354		LDX	TBPNTR	
3355		LDAA	0,X	GET BUFF CHAR

3356	INX		
3357	CPX	TXUVAL	ADJ BUFF PNTRS
3358	BNE	SOFLC	
3359	LDX	TXLVAL	
3360	SOFLC	STX	TBPNTR
3361	CPX	TXLGDP	LAST GOOD DATA?
3362	BNE	SOFLD	
3363	LDX	#0	YES
3364	STX	TXLGDP	
3365	STAB	TXDONE	
3366	SOFLD	STAA	PIA1BD
3367	LDX	OUTCHC	OUTPUT CHAR #10
3368	DEX		
3369	BNE	SOFM	
3370	JMP	SOFT	
3371	SOFM	STX	OUTCHC
3372	LDAA	DATA5	FIFO READY?
3373	BPL	SOFMA	
3374	JMP	SOFS	
3375	SOFMA	LDX	TXCHCT
3376	BNE	SOFMB	
3377	LDAA	#0FFH	GET STUFF CHAR
3378	STAA	TXEND	
3379	LDX	TXPNTR	
3380	STAA	13,X	SET PORT'S OUFG
3381	BRA	SOFMD	
3382	SOFMB	DEX	
3383	STX	TXCHCT	
3384	LDX	TBPNTR	
3385	LDAA	0,X	GET BUFF CHAR
3386	INX		
3387	CPX	TXUVAL	ADJ BUFF PNTRS
3388	BNE	SOFMC	
3389	LDX	TXLVAL	
3390	SOFMC	STX	TBPNTR
3391	CPX	TXLGDP	LAST GOOD DATA?
3392	BNE	SOFMD	
3393	LDX	#0	YES
3394	STX	TXLGDP	
3395	STAB	TXDONE	
3396	SOFMD	STAA	PIA1BD
3397	LDX	OUTCHC	OUTPUT CHAR #11
3398	DEX		
3399	BNE	SOFN	
3400	JMP	SOFT	
3401	SOFN	STX	OUTCHC
3402	LDAA	DATA5	FIFO READY?
3403	BPL	SOFNA	
3404	JMP	SOFS	
3405	SOFNA	LDX	TXCHCT
3406	BNE	SOFNB	
3407	LDAA	#0FFH	GET STUFF CHAR
3408	STAA	TXEND	
3409	LDX	TXPNTR	
3410	STAA	13,X	SET PORT' OUFG
3411	BRA	SOFND	
3412	SOFNB	DEX	

3413	STX	TXCHCT	
3414	LDX	TBPNTR	
3415	LDAA	0,X	GET BUFF CHAR
3416	INX		
3417	CPX	TXUVAL	AND BUFF PNTRS
3418	BNE	SOFNC	
3419	LDX	TXLVAL	
3420	SOFNC	STX	TBPNTR
3421	CPX	TXLGDP	LAST GOOD DATA?
3422	BNE	SOFND	
3423	LDX	#0	YES
3424	STX	TXLGDP	
3425	STAB	TXDONE	
3426	SOFND	STAA	PIA1BD
3427	LDX	OUTCHC	OUTPUT CHAR #12
3428	DEX		
3429	BNE	SOFO	
3430	JMP	SOFT	
3431	SOFO	STX	OUTCHC
3432	LDAA	DATA5	FIFO READY?
3433	BPL	SOFOA	
3434	JMP	SOFS	
3435	SOFOA	LDX	TXCHCT
3436	BNE	SOF0B	
3437	LDAA	#0FFH	GET STUFF CHAR
3438	STAA	TXEND	
3439	LDX	TXPNTR	
3440	STAA	13,X	SET PORT'S OUF0
3441	BRA	SOF0D	
3442	SOF0B	DEX	
3443	STX	TXCHCT	
3444	LDX	TBPNTR	
3445	LDAA	0,X	GET BUFF CHAR
3446	INX		
3447	CPX	TXUVAL	ADJ BUFF PNTRS
3448	BNE	SOF0C	
3449	LDX	TXLVAL	
3450	SOF0C	STX	TBPNTR
3451	CPX	TXLGDP	LAST GOOD DATA?
3452	BNE	SOF0D	
3453	LDX	#0	YES
3454	STX	TXLGDP	
3455	STAB	TXDONE	
3456	SOF0D	STAA	PIA1BD
3457	LDX	OUTCHC	OUTPUT CHAR #13
3458	DEX		
3459	BNE	SOF0P	
3460	JMP	SOFT	
3461	SOF0P	STX	OUTCHC
3462	LDAA	DATA5	FIFO READY?
3463	BPL	SOF0A	
3464	JMP	SOFS	
3465	SOF0A	LDX	TXCHCT
3466	BNE	SOF0B	
3467	LDAA	#0FFH	GET BUFF CHAR
3468	STAA	TXEND	
3469	LDX	TXPNTR	

3470	STAA	13,X	SET PORT'S OUFG
3471	BRA	SOFPD	
3472	SOFPB	DEX	
3473	STX	TXCHCT	
3474	LDX	TBPNTR	
3475	LDAA	0,X	GET BUF CHAR
3476	INX		
3477	CPX	TXUVAL	ADJ BUFF PNTRS
3478	BNE	SOFPC	
3479	LDX	TXLVAL	
3480	SOFPC	STX	TBPNTR
3481	CPX	TXLGDP	LAST GOOD DATA?
3482	BNE	SOFPD	
3483	LDX	#0	YES
3484	STX	TXLGDP	
3485	STAB	TXDONE	
3486	SOFPD	STAA	PIA1BD
3487	LDX	OUTCHC	OUTPUT CHAR #14
3488	DEX		
3489	BNE	SOFQ	
3490	JMP	SOFT	
3491	SOFQ	STX	OUTCHC
3492	LDAA	DATA5	FIFO READY?
3493	BPL	SOFQA	
3494	JMP	SOFS	
3495	SOFQA	LDX	TXCHCT
3496	BNE	SOFQB	
3497	LDAA	#0FFH	GET STUFF CHAR
3498	STAA	TXEND	
3499	LDX	TXPNTR	
3500	STAA	13,X	SET PORT'S OUFG
3501	BRA	SOFQD	
3502	SOFQB	DEX	
3503	STX	TXCHCT	
3504	LDX	TBPNTR	
3505	LDAA	0,X	GET BUFF CHAR
3506	INX		
3507	CPX	TXUVAL	ADJ BUFF PNTRS
3508	BNE	SOFQC	
3509	LDX	TXLVAL	
3510	SOFQC	STX	TBPNTR
3511	CPX	TXLGDP	LAST GOOD DATA?
3512	BNE	SOFQD	
3513	LDX	#0	YES
3514	STX	TXLGDP	
3515	STAB	TXDONE	
3516	SOFQD	STAA	PIA1BD
3517	LDX	OUTCHC	OUTPUT CHAR #15
3518	DEX		
3519	BEQ	SOFT	
3520	SOFR	STX	OUTCHC
3521	LDAA	DATA5	FIFO READY?
3522	BPL	SOFRA	
3523	JMP	SOFS	
3524	SOFRA	LDX	TXCHCT
3525	BNE	SOFRB	
3526	LDAA	#0FFH	GET STUFF CHAR

3527	STAA	TXEND	
3528	LDX	TXPNTR	
3529	STAA	13,X	SET PORT'S OUFG
3530	BRA	SOFRD	
3531	SOFRB	DEX	
3532	STX	TXCHCT	
3533	LDX	TBPNTR	
3534	LDAA	0,X	GET BUFF CHAR
3535	INX		
3536	CPX	TXUVAL	ADJ BUFF PNTRS
3537	BNE	SOFRC	
3538	LDX	TXLVAL	
3539	SOFRC	STX	TBPNTR
3540	CPX	TXLGDP	LAST GOOD DATA?
3541	BNE	SOFRD	
3542	LDX	#0	YES
3543	STX	TXLGDP	
3544	STAB	TXDONE	
3545	SOFRD	STAA	PIA1BD
3546	LDX	OUTCHC	OUTPUT CHAR #16
3547	DEX		
3548	BEQ	SOFT	
3549	STX	OUTCHC	
3550	SOFS	LDX	TXPNTR
3551	LDAA	TXCHCT	STORE BUFF PNTRS
3552	LDAB	TXCHCT+1	
3553	STAA	4,X	RESTORE BUFF CHAR COUNT
3554	STAB	5,X	
3555	LDAA	TBPNTR	
3556	LDAB	TBPNTR+1	
3557	STAA	2,X	RESTORE BUFF OUTPUT PNTR
3558	STAB	3,X	
3559	LDAA	TXLGDP	
3560	LDAB	TXLGDP+1	
3561	STAA	14,X	RESTORE L.G.D. PNTR
3562	STAB	15,X	
3563	RTS		
3564	SOFT	LDAA	TXEND
3565	BEQ	SOFTB	
3566	SOFTA	CLRB	
3567	STAB	TXEND	
3568	STAB	TXDONE	
3569	STAB	TXCHCT	
3570	STAB	TXCHCT+1	
3571	LDAA	XMASK	CLR "DATA RDY PORT-"
3572	ANDA	PA1AD	
3573	STAA	PA1AD	
3574	STAA	PIA1AD	
3575	LDX	TXPNTR	
3576	STAB	8,X	CLR TX REQ FLAG
3577	STAB	13,X	CLR OUFG
3578	STAB	6,X	CLR EMPTY
3579	STAB	4,X	CLR CHCT
3580	STAB	5,X	
3581	LDAB	#3	
3582	STAB	7,X	SET FLBFG
3583	LDAA	MASKR	TX ACK OF I/O PORT

3584	ANDA	MRFTA	AND SET FIFO'S MR
3585	LDAB	XMASK	
3586	COMB		
3587	STAB	XMASK	
3588	ORAA	XMASK	
3589	STAA	MRFTA	
3590	STAA	LATCH3	
3591	BRA	SOFU	
3592	SOFTB	LDAA	TXDONE
3593	BEQ	SOFU	
3594	LDAA	XMASK	TEST PORT'S TX REQ INPUT
3595	COMA		
3596	ANDA	DATA1	
3597	BEQ	SOFTA	
3598	SOFU	CLRA	
3599	STAA	SOEN	
3600	BRA	SOFS	
3601	SOFY	LDAB	#0FFH
3602	LDAA	TXEND	
3603	BNE	SOFYA	
3604	LDAA	TXDONE	
3605	BEQ	SOFYXX	
3606	LDAA	XMASK	TEST PORT'S TX REQ INPUT
3607	COMA		
3608	ANDA	DATA1	
3609	BEQ	SOFYXX	BRANCH IF TX REQ = 0
3610	CLRA		GET HERE WITH MULTIPLE
3611	STAA	TXDONE	TX REQS. PER FRAME
3612	SOFYXX	JMP	SOFC
3613	SOFYA	LDX	OUTCHC
3614	LDAA	DATA5	FIFO RDY?
3615	BPL	SOFYA1	
3616	RTS		
3617	SOFYA1	STAB	PIA1BD OUTPUT FILL CHAR #1
3618	DEX		
3619	BEQ	SOFTA	
3620	SOFYB	LDAA	DATA5 FIFO RDY?
3621	BPL	SOFYB1	
3622	STX	OUTCHC	
3623	RTS		
3624	SOFYB1	STAB	PIA1BD OUTPUT FILL CHAR #2
3625	DEX		
3626	BEQ	SOFTA	
3627	SOFYC	LDAA	DATA5 FIFO RDY?
3628	BPL	SOFYC1	
3629	STX	OUTCHC	
3630	RTS		
3631	SOFYC1	STAB	PIA1BD OUTPUT FILL CHAR #3
3632	DEX		
3633	BNE	SOFYD	
3634	JMP	SOFTA	
3635	SOFYD	LDAA	DATA5 FIFO RDY?
3636	BPL	SOFYD1	
3637	STX	OUTCHC	
3638	RTS		
3639	SOFYD1	STAB	PIA1BD OUTPUT FILL CHAR #4
3640	DEX		

3641	BNE	SOFYE	
3642	JMP	SOFTA	
3643	SOFYE LDAA	DATA5	FIFO RDY?
3644	BPL	SOFYE1	
3645	STX	OUTCHC	
3646	RTS		
3647	SOFYE1 STAB	PIA1BD	OUTPUT FILL CHAR #5
3648	DEX		
3649	BNE	SOFYF	
3650	JMP	SOFTA	
3651	SOFYF LDAA	DATA5	FIFO RDY?
3652	BPL	SOFYF1	
3653	STX	OUTCHC	
3654	RTS		
3655	SOFYF1 STAB	PIA1BD	OUTPUT FILL CHAR #6
3656	DEX		
3657	BNE	SOFYG	
3658	JMP	SOFTA	
3659	SOFYG LDAA	DATA5	FIFO RDY?
3660	BPL	SOFYG1	
3661	STX	OUTCHC	
3662	RTS		
3663	SOFYG1 STAB	PIA1BD	OUTPUT FILL CHAR #7
3664	DEX		
3665	BNE	SOFYH	
3666	JMP	SOFTA	
3667	SOFYH LDAA	DATA5	FIFO RDY?
3668	BPL	SOFYH1	
3669	STX	OUTCHC	
3670	RTS		
3671	SOFYH1 STAB	PIA1BD	OUTPUT FILL CHAR #8
3672	DEX		
3673	BNE	SOFYI	
3674	JMP	SOFTA	
3675	SOFYI LDAA	DATA5	FIFO RDY?
3676	BPL	SOFYI1	
3677	STX	OUTCHC	
3678	RTS		
3679	SOFYI1 STAB	PIA1BD	OUTPUT FILL CHAR #9
3680	DEX		
3681	BNE	SOFYJ	
3682	JMP	SOFTA	
3683	SOFYJ LDAA	DATA5	FIFO RDY?
3684	BPL	SOFYJ1	
3685	STX	OUTCHC	
3686	RTS		
3687	SOFYJ1 STAB	PIA1BD	OUTPUT FILL CHAR #10
3688	DEX		
3689	BNE	SOFYK	
3690	JMP	SOFTA	
3691	SOFYK LDAA	DATA5	FIFO RDY?
3692	BPL	SOFYK1	
3693	STX	OUTCHC	
3694	RTS		
3695	SOFYK1 STAB	PIA1BD	OUTPUT FILL CHAR #11
3696	DEX		
3697	BNE	SOFYL	

```

3698      JMP      SOFTA
3699 SOFYL  LDAA    DATA5      FIFO RDY?
3700      BPL      SOFYL1
3701      STX      OUTCHC
3702      RTS
3703 SOFYL1 STAB    PIA1BD      OUTPUT FILL CHAR #12
3704      DEX
3705      BNE      SOFYM
3706      JMP      SOFTA
3707 SOFYM  LDAA    DATA5      FIFO RDY?
3708      BPL      SOFYM1
3709      STX      OUTCHC
3710      RTS
3711 SOFYM1 STAB    PIA1BD      OUTPUT FILL CHAR #13
3712      DEX
3713      BNE      SOFYN
3714      JMP      SOFTA
3715 SOFYN  LDAA    DATA5      FIFO RDY?
3716      BPL      SOFYN1
3717      STX      OUTCHC
3718      RTS
3719 SOFYN1 STAB    PIA1BD      OUTPUT FILL CHAR #14
3720      DEX
3721      BNE      SOFYO
3722      JMP      SOFTA
3723 SOFYO  LDAA    DATA5      FIFO RDY?
3724      BPL      SOFYO1
3725      STX      OUTCHC
3726      RTS
3727 SOFYO1 STAB    PIA1BD      OUTPUT FILL CHAR #15
3728      DEX
3729      BNE      SOFYP
3730      JMP      SOFTA
3731 SOFYP  LDAA    DATA5      FIFO RDY?
3732      BPL      SOFYP1
3733      STX      OUTCHC
3734      RTS
3735 SOFYP1 STAB    PIA1BD      OUTPUT FILL CHAR #16
3736      DEX
3737      BNE      SOFYQ
3738      JMP      SOFTA
3739 SOFYQ  STX      OUTCHC
3740      RTS
3741 *****
3742 *
3743 *      BEGIN SERVICING SYSTEM INPUT FIFO
3744 *
3745 *
3746 SIF    LDAA    SIEN
3747      BNE      SIFA
3748      JMP      TDMA
3749 *
3750 SIF1   LDX      INCHC      *RX SELF TEST ROUTINE*
3751      LDAB      #16
3752 SIF2   LDAA    DATA1      CHAR IN RX FIFO?
3753      BPL      SIF6
3754      LDAA    DATA2      LOAD CHAR

```



```

3755      CMPA    #033H
3756      BEQ     SIF3      CORRECT CHARACTER?
3757      STAB    STERR
3758 SIF3    DEX
3759      BNE     SIF5
3760      CLRA
3761      STAA    SIEN
3762      STAA    SELFTR
3763      STX     INCHC
3764      LDAA    PA1AD
3765      LDAB    STERR
3766      BNE     SIF3A
3767      ANDA    #07FH      CLR DBP SELF TEST FAIL
3768      BRA     SIF4
3769 SIF3A   ORAA    #080H      SET DBP SELF TEST FAIL
3770 SIF4    STAA    PA1AD
3771      STAA    PIA1AD
3772      ANDA    #080H      ADJUST SELF TEST
3773      ORAA    BITELT     FAIL BITE LITE
3774      STAA    BITELT
3775      STAA    LATCH2
3776      CLR     STERR
3777      JMP     TDMA
3778 SIF5    DECB
3779      BNE     SIF2
3780 SIF6    STX     INCHC
3781      JMP     TDMA
3782 *
3783 SIFA    LDAA    SELFTR   RX SELF TEST?
3784      BNE     SIF1
3785      LDAA    INPTRC
3786      BEQ     SIFA1A
3787      LDX     RCRXPT
3788 SIFR    LDAA    DATA1   CHAR IN INPUT FIFO?
3789      BMI     SIFRA
3790 SIFR1   STX     RCRXPT
3791      JMP     TDMA
3792 SIFRA    LDAA    DATA2   LOAD & STORE FIFO CHAR
3793      STAA    0,X
3794      INX
3795      CPX     #ERCRX      BUFFER FULL?
3796      BNE     SIFR
3797      SEI
3798      LDX     #SRCRX
3799      LDAA    0,X
3800      STAA    ACIAD      LOAD ACIA WITH 1
3801      INX              R/CCOW RX CHARS.
3802      LDAA    #0BDH
3803      STAA    ACIACS
3804      STAA    TXIRFG
3805      STX     RCRXPT
3806      CLR     SIEN
3807      CLI
3808      JMP     TDMA
3809 *****
3810 *
3811 *FILL MISSING RX BURST WITH "FF" DATA

```

3812	*			
3813	SIFRB	LDAB	#16	INIT STORE COUNTER
3814	SIFRC	LDAA	#0FFH	
3815		LDX	RBPNTR	
3816		STAA	0,X	
3817		INX		ADJ PNTRS
3818		CPX	RXUVAL	
3819		BNE	SIFRD	
3820		LDX	RXLVAL	
3821	SIFRD	STX	RBPNTR	
3822		LDX	RBCHCT	
3823		INX		
3824		STX	RBCHCT	
3825		LDX	INCHC	TEST FIFO CH CNT
3826		DEX		
3827		BNE	SIFRE	
3828		CLR	FILLFG	
3829		JMP	SIFQ	
3830	SIFRE	STX	INCHC	
3831		DECB		DEC STORE COUNTER
3832		BNE	SIFRC	
3833		JMP	SIFP	
3834	*			
3835	*			
3836	SIFA1A	LDX	RXPNTR	SETUP RCV PNTRS
3837		LDAA	8,X	TEST PORT'S TX REQ FLAG
3838		BNE	SIFA1C	
3839		LDAA	PTONRX	
3840		ANDA	MASKIO	TEST IF PORT IS ON
3841		BNE	SIFA1B	
3842	SIFA1C	CLR	SIEN	*ABORT RX BURST: TX REQ*
3843		CLR	FILLFG	*OR PORT IS OFF*
3844		JMP	TDMA	
3845	SIFA1B	LDX	0,X	
3846		STX	RBPNTR	
3847		LDX	RXPNTR	
3848		LDX	4,X	
3849		STX	RBCHCT	
3850		LDAA	FILLFG	FILL BURST WITH "FF"?
3851		BNE	SIFRB	
3852		LDAA	DATA1	
3853		BMI	SIFA1	
3854		JMP	TDMA	
3855	SIFA1	LDAA	DATA2	LOAD & STORE CHAR #1
3856		LDX	RBPNTR	
3857		STAA	0,X	
3858		INX		ADJ PNTRS
3859		CPX	RXUVAL	
3860		BNE	SIFA2	
3861		LDX	RXLVAL	
3862	SIFA2	STX	RBPNTR	
3863		LDX	RBCHCT	
3864		INX		
3865		STX	RBCHCT	
3866		LDX	INCHC	TEST FIFO CH CNT
3867		DEX		
3868		BNE	SIFB	

3869	JMP	SIFQ	
3870	SIFB	STX	INCHC
3871		LDAA	DATA1
3872		BMI	SIFB1
3873		JMP	SIFP
3874	SIFB1	LDAA	DATA2
			LOAD & STORE FIFO CHAR #2
3875		LDX	RBPNTR
3876		STAA	0,X
3877		INX	ADJ PNTRS
3878		CPX	RXUVAL
3879		BNE	SIFB2
3880		LDX	RXLVAL
3881	SIFB2	STX	RBPNTR
3882		LDX	RBCHCT
3883		INX	
3884		STX	RBCHCT
3885		LDX	INCHC
			TEST FIFO CHAR COUNT
3886		DEX	
3887		BNE	SIFC
3888		JMP	SIFQ
3889	SIFC	STX	INCHC
3890		LDAA	DATA1
3891		BMI	SIFC1
3892		JMP	SIFP
3893	SIFC1	LDAA	DATA2
			LOAD & STORE FIFO CHAR #3
3894		LDX	RBPNTR
3895		STAA	0,X
3896		INX	ADJ PNTRS
3897		CPX	RXUVAL
3898		BNE	SIFC2
3899		LDX	RXLVAL
3900	SIFC2	STX	RBPNTR
3901		LDX	RBCHCT
3902		INX	
3903		STX	RBCHCT
3904		LDX	INCHC
			TEST FIFO CHAR COUNT
3905		DEX	
3906		BNE	SIFD
3907		JMP	SIFQ
3908	SIFD	STX	INCHC
3909		LDAA	DATA1
3910		BMI	SIFD1
3911		JMP	SIFP
3912	SIFD1	LDAA	DATA2
			LOAD & STORE FIFO CHAR #4
3913		LDX	RBPNTR
3914		STAA	0,X
3915		INX	ADJ PNTRS
3916		CPX	RXUVAL
3917		BNE	SIFD2
3918		LDX	RXLVAL
3919	SIFD2	STX	RBPNTR
3920		LDX	RBCHCT
3921		INX	
3922		STX	RBCHCT
3923		LDX	INCHC
			TEST FIFO CHAR COUNT
3924		DEX	
3925		BNE	SIFE

3926		JMP	SIFQ	
3927	SIFE	STX	INCHC	
3928		LDAA	DATA1	
3929		BMI	SIFE1	
3930		JMP	SIFP	
3931	SIFE1	LDAA	DATA2	LOAD & STORE FIFO CHAR #5
3932		LDX	RBPNTR	
3933		STAA	0,X	
3934		INX		ADJ PNTRS
3935		CPX	RXUVAL	
3936		BNE	SIFE2	
3937		LDX	RXLVAL	
3938	SIFE2	STX	RBPNTR	
3939		LDX	RBCHCT	
3940		INX		
3941		STX	RBCHCT	
3942		LDX	INCHC	TEST FIFO CHAR COUNT
3943		DEX		
3944		BNE	SIFF	
3945		JMP	SIFQ	
3946	SIFF	STX	INCHC	
3947		LDAA	DATA1	
3948		BMI	SIFF1	
3949		JMP	SIFP	
3950	SIFF1	LDAA	DATA2	LOAD & STORE FIFO CHAR #6
3951		LDX	RBPNTR	
3952		STAA	0,X	
3953		INX		ADJ PNTRS
3954		CPX	RXUVAL	
3955		BNE	SIFF2	
3956		LDX	RXLVAL	
3957	SIFF2	STX	RBPNTR	
3958		LDX	RBCHCT	
3959		INX		
3960		STX	RBCHCT	
3961		LDX	INCHC	TEST FIFO CHAR COUNT
3962		DEX		
3963		BNE	SIFG	
3964		JMP	SIFQ	
3965	SIFG	STX	INCHC	
3966		LDAA	DATA1	
3967		BMI	SIFG1	
3968		JMP	SIFP	
3969	SIFG1	LDAA	DATA2	LOAD & STORE FIFO CHAR #7
3970		LDX	RBPNTR	
3971		STAA	0,X	
3972		INX		ADJ PNTRS
3973		CPX	RXUVAL	
3974		BNE	SIFG2	
3975		LDX	RXLVAL	
3976	SIFG2	STX	RBPNTR	
3977		LDX	RBCHCT	
3978		INX		
3979		STX	RBCHCT	
3980		LDX	INCHC	TEST FIFO CHAR COUNT
3981		DEX		
3982		BNE	SIFH	

3983		JMP	SIFQ	
3984	SIFH	STX	INCHC	
3985		LDAA	DATA1	
3986		BMI	SIFH1	
3987		JMP	SIFP	
3988	SIFH1	LDAA	DATA2	LOAD & STORE FIFO CHAR #8
3989		LDX	RBPNTNTR	
3990		STAA	0,X	
3991		INX		ADJ PNTRS
3992		CPX	RXUVAL	
3993		BNE	SIFH2	
3994		LDX	RXLVAL	
3995	SIFH2	STX	RBPNTNTR	
3996		LDX	RBCHCT	
3997		INX		
3998		STX	RBCHCT	
3999		LDX	INCHC	TEST FIFO CHAR COUNT
4000		DEX		
4001		BNE	SIFI	
4002		JMP	SIFQ	
4003	SIFI	STX	INCHC	
4004		LDAA	DATA1	
4005		BMI	SIFI1	
4006		JMP	SIFP	
4007	SIFI1	LDAA	DATA2	LOAD & STORE FIFO CHAR #9
4008		LDX	RBPNTNTR	
4009		STAA	0,X	
4010		INX		ADJ PNTRS
4011		CPX	RXUVAL	
4012		BNE	SIFI2	
4013		LDX	RXLVAL	
4014	SIFI2	STX	RBPNTNTR	
4015		LDX	RBCHCT	
4016		INX		
4017		STX	RBCHCT	
4018		LDX	INCHC	TEST FIFO CHAR COUNT
4019		DEX		
4020		BNE	SIFX	
4021		JMP	SIFQ	
4022	SIFX	STX	INCHC	
4023		LDAA	DATA1	
4024		BMI	SIFX1	
4025		JMP	SIFP	
4026	SIFX1	LDAA	DATA2	LOAD & STORE FIFO CHAR #10
4027		LDX	RBPNTNTR	
4028		STAA	0,X	
4029		INX		ADJ PNTRS
4030		CPX	RXUVAL	
4031		BNE	SIFX2	
4032		LDX	RXLVAL	
4033	SIFX2	STX	RBPNTNTR	
4034		LDX	RBCHCT	
4035		INX		
4036		STX	RBCHCT	
4037		LDX	INCHC	TEST FIFO CHAR COUNT
4038		DEX		
4039		BNE	SIFJ	

4040		JMP	SIFQ	
4041	SIFJ	STX	INCHC	
4042		LDAA	DATA1	
4043		BMI	SIFJ1	
4044		JMP	SIFP	
4045	SIFJ1	LDAA	DATA2	LOAD & STORE FIFO CHAR #11
4046		LDX	RBPNTNTR	
4047		STAA	0,X	
4048		INX		
4049		CPX	RXUVAL	ADJ PNTRS
4050		BNE	SIFJ2	
4051		LDX	RXLVAL	
4052	SIFJ2	STX	RBPNTNTR	
4053		LDX	RBCHCT	
4054		INX		
4055		STX	RBCHCT	
4056		LDX	INCHC	TEST FIFO CHAR COUNT
4057		DEX		
4058		BNE	SIFK	
4059		JMP	SIFQ	
4060	SIFK	STX	INCHC	
4061		LDAA	DATA1	
4062		BMI	SIFK1	
4063		JMP	SIFP	
4064	SIFK1	LDAA	DATA2	LOAD & STORE FIFO CHAR #12
4065		LDX	RBPNTNTR	
4066		STAA	0,X	
4067		INX		ADJ PNTRS
4068		CPX	RXUVAL	
4069		BNE	SIFK2	
4070		LDX	RXLVAL	
4071	SIFK2	STX	RBPNTNTR	
4072		LDX	RBCHCT	
4073		INX		
4074		STX	RBCHCT	
4075		LDX	INCHC	TEST FIFO CHAR COUNT
4076		DEX		
4077		BNE	SIFL	
4078		JMP	SIFQ	
4079	SIFL	STX	INCHC	
4080		LDAA	DATA1	
4081		BMI	SIFL1	
4082		JMP	SIFP	
4083	SIFL1	LDAA	DATA2	LOAD & STORE FIFO CHAR #13
4084		LDX	RBPNTNTR	
4085		STAA	0,X	
4086		INX		ADJ PNTRS
4087		CPX	RXUVAL	
4088		BNE	SIFL2	
4089		LDX	RXLVAL	
4090	SIFL2	STX	RBPNTNTR	
4091		LDX	RBCHCT	
4092		INX		
4093		STX	RBCHCT	
4094		LDX	INCHC	TEST FIFO CHAR COUNT
4095		DEX		
4096		BNE	SIFM	

4097		JMP	SIFQ	
4098	SIFM	STX	INCHC	
4099		LDAA	DATA1	
4100		BPL	SIFP	
4101	SIFM1	LDAA	DATA2	LOAD & STORE FIFO CHAR #14
4102		LDX	RBPNTN	
4103		STAA	0,X	
4104		INX		ADJ PNTRS
4105		CPX	RXUVAL	
4106		BNE	SIFM2	
4107		LDX	RXLVAL	
4108	SIFM2	STX	RBPNTN	
4109		LDX	RBCHCT	
4110		INX		
4111		STX	RBCHCT	
4112		LDX	INCHC	TEST FIFO CHAR COUNT
4113		DEX		
4114		BEQ	SIFQ	
4115	SIFN	STX	INCHC	
4116		LDAA	DATA1	
4117		BPL	SIFP	
4118	SIFN1	LDAA	DATA2	LOAD & STORE FIFO CHAR #15
4119		LDX	RBPNTN	
4120		STAA	0,X	
4121		INX		ADJ PNTRS
4122		CPX	RXUVAL	
4123		BNE	SIFN2	
4124		LDX	RXLVAL	
4125	SIFN2	STX	RBPNTN	
4126		LDX	RBCHCT	
4127		INX		
4128		STX	RBCHCT	
4129		LDX	INCHC	TEST FIFO CHAR COUNT
4130		DEX		
4131		BEQ	SIFQ	
4132	SIFO	STX	INCHC	
4133		LDAA	DATA1	
4134		BPL	SIFP	
4135	SIFO1	LDAA	DATA2	LOAD & STORE FIFO CHAR #16
4136		LDX	RBPNTN	
4137		STAA	0,X	
4138		INX		ADJ PNTRS
4139		CPX	RXUVAL	
4140		BNE	SIFO2	
4141		LDX	RXLVAL	
4142	SIFO2	STX	RBPNTN	
4143		LDX	RBCHCT	
4144		INX		
4145		STX	RBCHCT	
4146		LDX	INCHC	TEST FIFO CHAR COUNT
4147		DEX		
4148		BEQ	SIFQ	
4149		STX	INCHC	
4150	SIFP	LDX	RXPNTN	UPDATE PORT'S BUFF PNTRS.
4151		LDAA	RBCHCT	
4152		LDAB	RBCHCT+1	
4153		STAA	4,X	

```

4154      STAB    5,X
4155      LDAA    RBPNTR
4156      LDAB    RBPNTR+1
4157      STAA    0,X
4158      STAB    1,X
4159      JMP     TDMA
4160 SIFQ      CLRA
4161      STAA    SIEN
4162      BRA     SIFP
4163 *****
4164 *
4165 *   INTERRUPT REQUEST SERVICE ROUTINE
4166 *
4167 *
4168 IRS      LDAB    ACIACS
4169      LDAA    TXIRFG
4170      BEQ     IRSC
4171      BITB    #2          TDR EMPTY?
4172      BEQ     IRSC
4173      LDX     RCRXPT
4174 IRSA      LDAA    0,X
4175      STAA    ACIAD      OUTPUT CHAR TO ACIA
4176      INX
4177      STX     RCRXPT
4178      CPX     #ERCRX
4179      BNE     IRSB
4180      LDAA    #09DH      CLR TX IRQ ENABLE
4181      STAA    ACIACS
4182      CLR     TXIRFG
4183 IRSB      BITB    #1          RDR FULL?
4184      BNE     IRSE
4185      RTI
4186 IRSC      BITB    #1
4187      BNE     IRSE
4188 IRSD      LDAA    #1          RCCOW/CCOW BITE ERROR:
4189      STAA    IRSBIT      IRQ ERROR OR SCU FORMAT ERR
4190      RTI
4191 IRSE      BITB    #070H      ACIA RX ERRORS?
4192      BEQ     IRSF
4193      LDAA    #1
4194      STAA    IRSBIT
4195 IRSF      LDAA    ACIAD      LOAD ACIA RX CHAR
4196      LDAB    RCFG
4197      BEQ     IRSJ
4198      DECB
4199      BEQ     IRSG
4200      DECB
4201      BNE     IRSN
4202      LDX     RTXPT
4203      STAA    0,X          STORE CHAR IN RCCOW BUFFER
4204      INX
4205      STX     RTXPT
4206      CPX     #ERTXB
4207      BNE     IRSI
4208      LDAA    #1          SET FOR RCCOW DATA READY
4209      STAA    RDRDY
4210      BRA     IRSH

```


4211	IRSG	LDX	CTXPT		
4212		STAA	0,X	STORE CHAR IN CCOW BUFFER	
4213		INX			
4214		STX	CTXPT		
4215		CPX	#ECTXB		
4216		BNE	IRSI		
4217		LDAA	#2	SET FOR CCOW DATA RDY	
4218		STAA	CDRDY		
4219	IRSH	CLR	RCFG		
4220	IRSI	RTI			
4221	IRSJ	CMPA	#0A3H		
4222		BNE	IRSL		
4223		LDX	#SCTXB		
4224		STX	CTXPT		
4225		LDAB	#1		
4226	IRSK	STAB	RCFG		
4227		RTI			
4228	IRSL	CMPA	#0A4H		
4229		BNE	IRSM		
4230		LDX	#SRTXB		
4231		STX	RTXPT		
4232		LDAB	#2		
4233		BRA	IRSK		
4234	IRSM	CMPA	#0A9H		
4235		BEQ	IRSMA		
4236		CMPA	#0ABH		WBR
4237		BEQ	IRSMA		WBR
4238		BRA	IRSD		WBR
4239	IRSMA	TAB		GET RXINHB BIT 2 FROM A9/A11 BIT 2	WBR
4240		ANDB	#02H		WBR
4241		STAB	RXINHB		WBR
4242		LDAB	#3		
4243		STAB	RC3FG		
4244		BRA	IRSK		
4245	IRSN	LDAB	RC3FG		
4246		CMPB	#3		
4247		BEQ	IRSQ		
4248		CMPB	#2		
4249		BEQ	IRSO		
4250		TAB		GET RXINHB BITS 4,3 FROM BITS 7,6	WBR
4251		RORB			WBR
4252		RORB			WBR
4253		RORB			WBR
4254		RORB			WBR
4255		ANDB	#0CH		WBR
4256		ORAB	RXINHB		WBR
4257		STAB	RXINHB		SEF
4258		ANDA	#03FH	REMOVE RXINHB BITS BEFORE USING	WBR
4259		STAA	BRATE2	PORT BIT RATES (3 & 4)	
4260		TAB			
4261		ANDB	#7		
4262		ANDA	#038H		
4263		ASLB			
4264		ASLB			
4265		ASLB			
4266		STAB	CHRTAT3		
4267		STAA	CHRTAT4		

```

4268      CLR      RCFG
4269      CLR      RC3FG
4270      RTI
4271  IRSO      TAB              GET RXINHB BIT 0 FROM BIT 6          WBR
4272      ROLB              WBR
4273      ROLB              WBR
4274      ROLB              WBR
4275      ANDB      #01H          WBR
4276      ORAB      RXINHB        WBR
4277      STAB      RXINHB        WBR
4278      ANDA      #0BFH      REMOVE RXINHB BIT BEFORE USING      SEF
4279      STAA      BRATE1      PORT BIT RATES (1 & 2)          WBR
4280      TAB
4281      ANDB      #080H              WBR
4282      STAB      BPS16K
4283      TAB
4284      ANDB      #7
4285      ANDA      #038H
4286      ASLB
4287      ASLB
4288      ASLB
4289      STAB      CHRAT1
4290      STAA      CHRAT2
4291  IRSP      DEC      RC3FG
4292      RTI
4293  IRSQ      STAA      PTONRX      PORT ON & RX ONLY INFO
4294      BRA      IRSP
4295  *****
4296  *
4297  *  LOAD RX ELASTIC BUFFER FOR 16 Kbps
4298  *
4299  STUFF      LDX      #SBUFF1
4300      CLRB
4301  STUFA      STAA      0,X
4302      INX
4303      INCB
4304      BNE      STUFA
4305      RTS
4306  *****
4307  *
4308  INSERR      BRA      INSERR      BAD INTERRUPT- CRASH
4309  *
4310  *****
4311  *
4312  *  PROGRAM VERSION, ROM CHECKSUM VALUES, AND INTERRUPT VECTORS.
4313  *
4314  *****
4315      ORG      0FFF5H
4316  PGMVRS      FCB      4          **PROGRAM VERSION**
4317  CKSMB      RMB      1          MSB OF CHECKSUM - BD. PROGRAMMER SUPPLIES VALUE
4318  CKSMA      RMB      1          LSB OF CHECKSUM - SEE 98-P08293T FOR DETAILS.
4319      FDB      IRS          IRQ VECTOR
4320      FDB      INSERR      SWI ERROR
4321      FDB      INSERR      NMI ERROR
4322      FDB      PWRON      RESTART VECTOR
4323  *
4324      END      PWRON

```

APPENDIX E

A. DATA BUFFER PROCESSOR LABEL CROSS REFERENCE

Label Name	Defined On	Referenced On							
BITRAT	642	637	943						
BSERV	2801	2351	2385	2406	2524	2528	2566	2580	2662
		2684	2720	2740	2748	2757	2761	2765	2768
		2771	2775	2779	2783	2787	2791		
BSERVA	2806	2803							
BSERVB	2821	2807							
BSERVC	2826	2809							
BSERVD	2832								
BSERVE	2833	2831	2835						
BSERVF	2834	2829							
BSERVG	2848	2843	2845						
BSERVH	2858	2853	2855						
BSERVI	2868	2863	2865						
BSERVK	2878	2873	2875						
CKSMA	4318								
CKSMB	4317								
CLEAR	116	29	126						
CLEARA	118	120							
INIT	68	11	175						
INSERR	4308	4308							
IO1A	1128	988							
IO1A1	989	986							
IO1A2	992	990							
IO1A3	1025	1023							
IO1AA	1132	1130							
IO1B	1136	1134							
IO1BA	1144	1147							
IO1BB	1145	1143							
IO1C	1148	1137	1139						
IO1CA	1152	1141							
IO1D	1160	1149	1314						
IO1E	1164	1151							
IO1F	1168	1166							
IO1G	1200	1342							
IO1H	1210	1161							
IO1HA	1214	1212							
IO1HB	1217	1215							
IO1I	1221	1219	1284						
IO1K	1226	1224							

Label Name	Defined On	Referenced On							
IOILF	1065	1387	1391	1395	1401	1405	1409	1413	1417
		1421	1425						
IOILF1	1071	1068							
IOILFA	1078	1073							
IOILFB	1081	1075	1077	1079					
IOILFC	1095	1085	1089						
IOILFD	1102	1098							
IOILS	1030	1221	1226	1231	1236	1244	1249	1254	1259
		1264	1269	1274					
IOILSA	1053	1048							
IOILSB	1056	1050	1052	1054					
IOILSX	1047	1037							
IOIM	1231	1229							
IOIN10	1274	1272							
IOIN2	1236	1234							
IOIN3	1240	1238							
IOIN4	1244	1242							
IOIN5	1249	1247							
IOIN6	1254	1252							
IOIN7	1259	1257							
IOIN8	1264	1262							
IOIN9	1269	1267							
IOIO	1276	1216							
IOIP	1279								
IOIPT	984	712	729	739	778	816	854	927	932
		976							
IOIXP	1285	1220	1225	1230	1235	1239	1243	1248	1253
		1258	1263	1268	1273	1275	1278		
IOIXQ	1291	1286							
IOIY	1292	1131	1135						
IOIYA	1314	1303	1316						
IOIYA1	1318	1213							
IOIYB	1315	1293							
IOIYC	1330								
IOIYC1	1338	1376	1384						
IOIYD	1343	1144	1317						
IOIYE	1350	1348							
IOIYE0	1369	1363	1373						
IOIYE1	1370	1101	1379						
IOIYF	1378	1344							
IOIYFA	1385	1382							
IOIYG	1387								
IOIYH	1391								
IOIYI	1395								
IOIYJ	1398								

Label Name	Defined On	Referenced On									
IO1YK	1401										
IO1YL	1405										
IO1YM	1409										
IO1YN	1413										
IO1YO	1417										
IO1YP	1421										
IO1YQ	1425										
IO2A	1574	1434									
IO2A1	1435	1432									
IO2A2	1438	1436									
IO2A3	1471	1469									
IO2AA	1578	1576									
IO2B	1582	1580									
IO2BA	1590	1593									
IO2BB	1591	1589									
IO2C	1594	1583	1585								
IO2CA	1598	1587									
IO2D	1606	1595	1771								
IO2DA	1609	1607									
IO2E	1611	1597									
IO2F	1615	1613									
IO2FA	1639	1634									
IO2FB	1642	1638									
IO2G	1653	1799									
IO2GA	1657	1654									
IO2GB	1658	1656									
IO2H	1667	1608									
IO2HA	1671	1669									
IO2HB	1674	1672									
IO2I	1678	1676	1741								
IO2K	1683	1681									
IO2LF	1511	1844	1848	1852	1858	1862	1866	1870	1874		
		1878	1882								
IO2LF1	1517	1514									
IO2LFA	1524	1519									
IO2LFB	1527	1521	1523	1525							
IO2LFC	1541	1531	1535								
IO2LFD	1548	1544									
IO2LS	1476	1678	1683	1688	1693	1701	1706	1711	1716		
		1721	1726	1731							
IO2LSA	1499	1494									
IO2LSB	1502	1496	1498	1500							
IO2LSX	1493	1483									
IO2M	1688	1686									
IO2N10	1731	1729									
IO2N2	1693	1691									

Label Name	Defined On	Referenced On							
IO2N3	1697	1695							
IO2N4	1701	1699							
IO2N5	1706	1704							
IO2N6	1711	1709							
IO2N7	1716	1714							
IO2N8	1721	1719							
IO2N9	1726	1724							
IO2O	1733	1673							
IO2P	1736								
IO2PT	1430	991	1025	1046	1163	1167	1199	1209	1291
		1313	1349	1369	1377	1386	1390	1394	1397
		1400	1404	1408	1412	1416	1420	1424	
IO2XP	1742	1677	1682	1687	1692	1696	1700	1705	1710
		1715	1720	1725	1730	1732	1735		
IO2XQ	1748	1743							
IO2Y	1749	1577	1581						
IO2YA	1771	1760	1773						
IO2YA1	1775	1670							
IO2YB	1772	1750							
IO2YC	1787								
IO2YC1	1795	1833	1841						
IO2YD	1800	1590	1774						
IO2YE	1807	1805							
IO2YE0	1826	1820	1830						
IO2YE1	1827	1547	1836						
IO2YF	1835	1801							
IO2YFA	1842	1839							
IO2YG	1844								
IO2YH	1848								
IO2YI	1852								
IO2YJ	1855								
IO2YK	1858								
IO2YL	1862								
IO2YM	1866								
IO2YN	1870								
IO2YO	1874								
IO2YP	1878								
IO2YQ	1882								
IO3A	2031	1891							
IO3A1	1892	1889							
IO3A2	1895	1893							
IO3A3	1928	1926							
IO3AA	2035	2033							
IO3B	2039	2037							
IO3BA	2047	2050							
IO3BB	2048	2046							

Label Name	Defined On	Referenced On							
IO3C	2051	2040	2042						
IO3CA	2055	2044							
IO3D	2063	2052	2228						
IO3DA	2066	2064							
IO3E	2068	2054							
IO3F	2072	2070							
IO3FA	2096	2091							
IO3FB	2099	2095							
IO3G	2110	2256							
IO3GA	2114	2111							
IO3GB	2115	2113							
IO3H	2124	2065							
IO3HA	2128	2126							
IO3HB	2131	2129							
IO3I	2135	2133	2198						
IO3K	2140	2138							
IO3LF	1968	2301	2305	2309	2315	2319	2323	2327	2331
		2335	2339						
IO3LFI	1974	1971							
IO3LFA	1981	1976							
IO3LFB	1984	1978	1980	1982					
IO3LFC	1998	1988	1992						
IO3LFD	2005	2001							
IO3LS	1933	2135	2140	2145	2150	2158	2163	2168	2173
		2178	2183	2188					
IO3LSA	1956	1951							
IO3LSB	1959	1953	1955	1957					
IO3LSX	1950	1940							
IO3M	2145	2143							
IO3N10	2188	2186							
IO3N2	2150	2148							
IO3N3	2154	2152							
IO3N4	2158	2156							
IO3N5	2163	2161							
IO3N6	2168	2166							
IO3N7	2173	2171							
IO3N8	2178	2176							
IO3N9	2183	2181							
IO3O	2190	2130							
IO3P	2193								
IO3PT	1887	1437	1471	1492	1610	1614	1652	1666	1748
		1770	1806	1826	1834	1843	1847	1851	1854
		1857	1861	1865	1869	1873	1877	1881	
IO3XP	2199	2134	2139	2144	2149	2153	2157	2162	2167
		2172	2177	2182	2187	2189	2192		
IO3XQ	2205	2200							

Label Name	Defined On	Referenced On	
IO3Y	2206	2034	2038
IO3YA	2228	2217	2230
IO3YA1	2232	2127	
IO3YB	2229	2207	
IO3YC	2244		
IO3YC1	2252	2290	2298
IO3YD	2257	2047	2231
IO3YE	2264	2262	
IO3YE0	2283	2277	2287
IO3YE1	2284	2004	2293
IO3YF	2292	2258	
IO3YFA	2299	2296	
IO3YG	2301		
IO3YH	2305		
IO3YI	2309		
IO3YJ	2312		
IO3YK	2315		
IO3YL	2319		
IO3YM	2323		
IO3YN	2327		
IO3YO	2331		
IO3YP	2335		
IO3YQ	2339		
IO4A	2488	2348	
IO4A1	2349	2346	
IO4A2	2352	2350	
IO4A3	2385	2383	
IO4AA	2492	2490	
IO4B	2496	2494	
IO4BA	2504	2507	
IO4BB	2505	2503	
IO4C	2508	2497	2499
IO4CA	2512	2501	
IO4D	2520	2509	2685
IO4DA	2523	2521	
IO4E	2525	2511	
IO4F	2529	2527	
IO4FA	2553	2548	
IO4FB	2556	2552	
IO4G	2567	2713	
IO4GA	2571	2568	
IO4GB	2572	2570	
IO4H	2581	2522	
IO4HA	2585	2583	
IO4HB	2588	2586	
IO4I	2592	2590	2655

Label Name	Defined On	Referenced On							
IO4K	2597	2595							
IO4LF	2425	2758	2762	2766	2772	2776	2780	2784	2788
		2792	2795						
IO4LF1	2431	2428							
IO4LFA	2438	2433							
IO4LFB	2441	2435	2437	2439					
IO4LFC	2455	2445	2449						
IO4LFD	2462	2458							
IO4LS	2390	2592	2597	2602	2607	2615	2620	2625	2630
		2635	2640	2645					
IO4LSA	2413	2408							
IO4LSB	2416	2410	2412	2414					
IO4LSX	2407	2397							
IO4M	2602	2600							
IO4N10	2645	2643							
IO4N2	2607	2605							
IO4N3	2611	2609							
IO4N4	2615	2613							
IO4N5	2620	2618							
IO4N6	2625	2623							
IO4N7	2630	2628							
IO4N8	2635	2633							
IO4N9	2640	2638							
IO4O	2647	2587							
IO4P	2650								
IO4PT	2344	1894	1928	1949	2067	2071	2109	2123	2205
		2227	2263	2283	2291	2300	2304	2308	2311
		2314	2318	2322	2326	2330	2334	2338	
IO4XP	2656	2591	2596	2601	2606	2610	2614	2619	2624
		2629	2634	2639	2644	2646	2649		
IO4XQ	2662	2657							
IO4Y	2663	2491	2495						
IO4YA	2685	2674	2687						
IO4YA1	2689	2584							
IO4YB	2686	2664							
IO4YC	2701								
IO4YC1	2709	2747	2755						
IO4YD	2714	2504	2688						
IO4YE	2721	2719							
IO4YE0	2740	2734	2744						
IO4YE1	2741	2461	2750						
IO4YF	2749	2715							
IO4YFA	2756	2753							
IO4YG	2758								
IO4YH	2762								
IO4YI	2766								

Label Name	Defined On	Referenced On							
IO4YJ	2769								
IO4YK	2772								
IO4YL	2776								
IO4YM	2780								
IO4YN	2784								
IO4YO	2788								
IO4YP	2792								
IO4YQ	2795								
IRS	4168								
IRSA	4174								
IRSB	4183	4179							
IRSC	4186	4170	4172						
IRSD	4188	4238							
IRSE	4191	4184	4187						
IRSF	4195	4192							
IRSG	4211	4199							
IRSH	4219	4210							
IRSI	4220	4207	4216						
IRSJ	4221	4197							
IRSK	4226	4233	4244						
IRSL	4228	4222							
IRSM	4234	4229							
IRSMA	4239	4235	4237						
IRSN	4245	4201							
IRSO	4271	4249							
IRSP	4291	4294							
IRSQ	4293	4247							
LD1	1107	1360	1361	1364	1365	1366	1367	1368	
LD2	1553	1817	1818	1821	1822	1823	1824	1825	
LD3	2010	2274	2275	2278	2279	2280	2281	2282	
LD4	2467	2731	2732	2735	2736	2737	2738	2739	
PGMVRS	4316								
PWRON	6								
PWRON1	58	30							
RAMERR	31	19	34						
RAMT	16	27							
RAMTST	17	22							
RBI	195	186	900						
RBIIC	225	222							
RBIICA	228	226							
RBIID	232	230							
RBIIDA	241	237							
RBIIE	242	234							
RBIIEA	252	247							
RBIIG	264	245							

Label Name	Defined On	Referenced On	
RBI1GA	271		
RBI1GB	284	278	
RBI1H	293	266	283
RBI1H1	294	263	
RBI1HA	303	297	
RBI1HB	307	302	
RBI2B	313	223	
RBI2C	316	314	
RBI2CA	319	317	
RBI2D	323	321	
RBI2DA	332	328	
RBI2E	333	325	
RBI2EA	343	338	
RBI2G	355	336	
RBI2GA	362		
RBI2GB	377	369	
RBI2H	386	357	376
RBI2H1	387	354	
RBI2HA	396	390	
RBI2HB	400	395	
RBI3B	406	315	
RBI3C	409	407	
RBI3CA	412	410	
RBI3D	416	414	
RBI3DA	425	421	
RBI3E	426	418	
RBI3EA	436	431	
RBI3G	448	429	
RBI3GA	455		
RBI3GB	470	462	
RBI3H	479	450	469
RBI3H1	480	447	
RBI3HA	489	483	
RBI3HB	493	488	
RBI4B	499	408	
RBI4C	502	500	
RBI4CA	505	503	
RBI4D	509	507	
RBI4DA	518	514	
RBI4E	519	511	
RBI4EA	529	524	
RBI4G	541	522	
RBI4GA	548		
RBI4GB	563	555	
RBI4H	572	543	562

Label Name	Defined On	Referenced On			
RBI4H1	573	540			
RBI4HA	582	576			
RBI4HB	586	581			
RBIA	198	196			
RBIB	210	199			
RBIBA	217	213			
RBIJ	594	501			
RBIJA	600	595			
RBIK	610	602			
RBIKA	616	634			
RBIL	619	614			
RBIM	625	636			
RBIN	631	612			
RBIO	635	632			
RBIS	637	311	404	497	590
RBISA	651	647			
RBISB	656	652			
RBISC	661	657			
RBISD	666	662			
RBISE	671	667			
RBISF	674	643			
RBIT	683	599	638		
RBIU	677	645			
ROMERR	53	48	50	56	
ROMTSA	41	46			
ROMTST	38	28			
SERVF1	2840	2847			
SERVF2	2844	2841			
SERVF3	2846	2839			
SERVG1	2850	2857			
SERVG2	2854	2851			
SERVG3	2856	2849			
SERVH1	2860	2867			
SERVH2	2864	2861			
SERVH3	2866	2859			
SERVH1	2870	2877			
SERVH2	2874	2871			
SERVH3	2876	2869			
SETUP	126				
SETUP1	127	58	188		
SETUP3	152	141			
SIF	3746	2886			
SIF1	3750	3784			
SIF2	3752	3779			
SIF3	3758	3756			

Label Name	Defined On	Referenced On
SIF3A	3769	3766
SIF4	3770	3768
SIF5	3778	3759
SIF6	3780	3753
SIFA	3783	3747
SIFA1	3855	3853
SIFA1A	3836	3786
SIFA1B	3845	3841
SIFA1C	3842	3838
SIFA2	3862	3860
SIFB	3870	3868
SIFB1	3874	3872
SIFB2	3881	3879
SIFC	3889	3887
SIFC1	3893	3891
SIFC2	3900	3898
SIFD	3908	3906
SIFD1	3912	3910
SIFD2	3919	3917
SIFE	3927	3925
SIFE1	3931	3929
SIFE2	3938	3936
SIFF	3946	3944
SIFF1	3950	3948
SIFF2	3957	3955
SIFG	3965	3963
SIFG1	3969	3967
SIFG2	3976	3974
SIFH	3984	3982
SIFH1	3988	3986
SIFH2	3995	3993
SIFI	4003	4001
SIFI1	4007	4005
SIFI2	4014	4012
SIFJ	4041	4039
SIFJ1	4045	4043
SIFJ2	4052	4050
SIFK	4060	4058
SIFK1	4064	4062
SIFK2	4071	4069
SIFL	4079	4077
SIFL1	4083	4081
SIFL2	4090	4088
SIFM	4098	4096
SIFM1	4101	

Label Name	Defined On	Referenced On							
SIFM2	4108	4106							
SIFN	4115								
SIFN1	4118								
SIFN2	4125	4123							
SIFO	4132								
SIFO1	4135								
SIFO2	4142	4140							
SIFP	4150	3833	3873	3892	3911	3930	3949	3968	3987
		4006	4025	4044	4063	4082	4100	4117	4134
		4162							
SIFQ	4160	3829	3869	3888	3907	3926	3945	3964	3983
		4002	4021	4040	4059	4078	4097	4114	4131
		4148							
SIFR	3788	3796							
SIFR1	3790								
SIFRA	3792	3789							
SIFRB	3813	3851							
SIFRC	3814	3832							
SIFRD	3821	3819							
SIFRE	3830	3827							
SIFX	4022	4020							
SIFX1	4026	4024							
SIFX2	4033	4031							
SOF	2883	2833							
SOF1	2886	2884							
SOF2	2888	2907							
SOF3	2890	2902							
SOF4	2901	2895							
SOF5	2903	2891							
SOFA	2906	2885							
SOFAA	2914	979	2911						
SOFB	2925	2923							
SOFB1	3048	3045							
SOFB2	3053	3050							
SOFB3	3057	3054							
SOFB4	3062	3059							
SOFB5	3066	3063							
SOFB6	3070	3068							
SOFBA	2931	2928							
SOFBB	2936	2933							
SOFBC	2940	2937							
SOFBD	2945	2942							
SOFBE	2949	2946							
SOFBF	2954	2951							
SOFBG	2958	2955							

Label Name	Defined On	Referenced On
SOFBH	2963	2960
SOFBI	2967	2964
SOFBJ	2972	2969
SOFBK	2976	2973
SOFBL	2981	2978
SOFBM	2985	2982
SOFBN	2990	2987
SOFBO	2994	2991
SOFBP	2999	2996
SOFBQ	3003	3000
SOFBR	3008	3005
SOFBS	3012	3009
SOFBT	3017	3014
SOFBU	3021	3018
SOFBV	3026	3023
SOFBW	3030	3027
SOFBX	3035	3032
SOFBY	3039	3036
SOFBZ	3044	3041
SOFC	3072	3612
SOFCA	3075	3073
SOFCB	3082	3076
SOFCC	3090	3088
SOFCD	3096	3081 3092
SOFD	3101	3099
SOFDA	3105	3103
SOFDB	3112	3106
SOFDC	3120	3118
SOFDD	3126	3111 3122
SOFE	3131	3129
SOFEA	3135	3133
SOFEB	3142	3136
SOFEC	3150	3148
SOFED	3156	3141 3152
SOFF	3161	3159
SOFFA	3165	3163
SOFFB	3172	3166
SOFFC	3180	3178
SOFFD	3186	3171 3182
SOFG	3191	3189
SOFGA	3195	3193
SOFGB	3202	3196
SOFGC	3210	3208
SOFGD	3216	3201 3212
SOFH	3221	3219

Label Name	Defined On	Referenced On
SOFHA	3225	3223
SOFHB	3232	3226
SOFHC	3240	3238
SOFHD	3246	3231 3242
SOFI	3251	3249
SOFIA	3255	3253
SOFIB	3262	3256
SOFIC	3270	3268
SOFID	3276	3261 3272
SOFJ	3281	3279
SOFJA	3285	3283
SOFJB	3292	3286
SOFJC	3300	3298
SOFJD	3306	3291 3302
SOFK	3311	3309
SOFKA	3315	3313
SOFKB	3322	3316
SOFKC	3330	3328
SOFKD	3336	3321 3332
SOFL	3341	3339
SOFLA	3345	3343
SOFLB	3352	3346
SOFLC	3360	3358
SOFLD	3366	3351 3362
SOFM	3371	3369
SOFMA	3375	3373
SOFMB	3382	3376
SOFMC	3390	3388
SOFMD	3396	3381 3392
SOFN	3401	3399
SOFNA	3405	3403
SOFNB	3412	3406
SOFNC	3420	3418
SOFND	3426	3411 3422
SOFO	3431	3429
SOFOA	3435	3433
SOFOB	3442	3436
SOFOC	3450	3448
SOFOD	3456	3441 3452
SOFP	3461	3459
SOFPA	3465	3463
SOFPB	3472	3466
SOFPC	3480	3478
SOFPD	3486	3471 3482
SOFQ	3491	3489

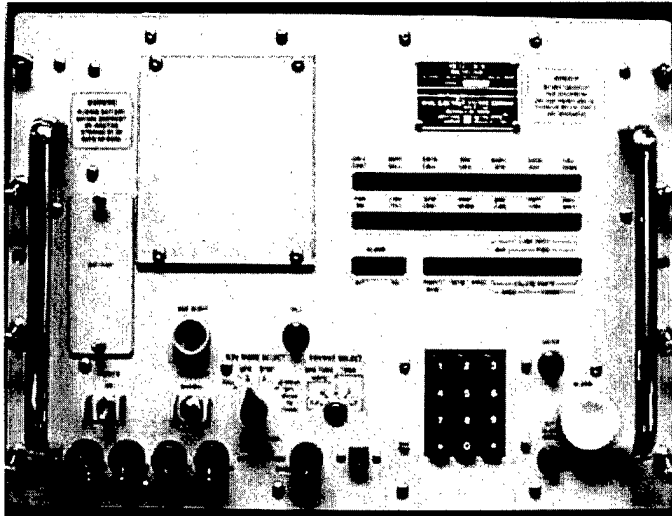
Label Name	Defined On	Referenced On							
SOFQA	3495	3493							
SOFQB	3502	3496							
SOFQC	3510	3508							
SOFQD	3516	3501	3512						
SOFR	3520								
SOFRA	3524	3522							
SOFRB	3531	3525							
SOFRC	3539	3537							
SOFRD	3545	3530	3541						
SOF S	3550	3074	3104	3134	3164	3194	3224	3254	3284
		3314	3344	3374	3404	3434	3464	3494	3523
		3600							
SOFT	3564	3100	3130	3160	3190	3220	3250	3280	3310
		3340	3370	3400	3430	3460	3490	3519	3548
SOFTA	3566	3597	3619	3626	3634	3642	3650	3658	3666
		3674	3682	3690	3698	3706	3714	3722	3730
		3738							
SOFTB	3592	3565							
SOFU	3598	3591	3593						
SOFY	3601	2924							
SOFYA	3613	3603							
SOFYA1	3617	3615							
SOFYB	3620								
SOFYB1	3624	3621							
SOFYC	3627								
SOFYC1	3631	3628							
SOFYD	3635	3633							
SOFYD1	3639	3636							
SOFYE	3643	3641							
SOFYE1	3647	3644							
SOFYF	3651	3649							
SOFYF1	3655	3652							
SOFYG	3659	3657							
SOFYG1	3663	3660							
SOFYH	3667	3665							
SOFYH1	3671	3668							
SOFYI	3675	3673							
SOFYI1	3679	3676							
SOFYJ	3683	3681							
SOFYJ1	3687	3684							
SOFYK	3691	3689							
SOFYK1	3695	3692							
SOFYL	3699	3697							
SOFYL1	3703	3700							
SOFYM	3707	3705							

Label Name	Defined On	Referenced On							
SOFYM1	3711	3708							
SOFYN	3715	3713							
SOFYN1	3719	3716							
SOFYO	3723	3721							
SOFYO1	3727	3724							
SOFYP	3731	3729							
SOFYP1	3735	3732							
SOFYQ	3739	3737							
SOFYXX	3612	3605	3609						
STUFA	4301	4304							
STUFF	4299	279	370	463	556				
TBI	692	197	227	231	251	318	322	342	411
		415	435	504	508	528	617	630	
TBI1	698	693							
TBI1BA	741	738							
TBI1E	748	742							
TBI1EA	752	747							
TBI1F	768	766							
TBI2	703	697							
TBI2B	773	735							
TBI2BA	779	777							
TBI2E	786	780							
TBI2EA	790	785							
TBI2F	806	804							
TBI3	705	699							
TBI3B	811	774							
TBI3BA	817	815							
TBI3E	824	818							
TBI3EA	828	823							
TBI3F	844	842							
TBI4	710	706							
TBI4B	849	812							
TBI4BA	855	853							
TBI4E	862	856							
TBI4EA	866	861							
TBI4F	882	880							
TBIA	713	711							
TBIB	721	714							
TBIBA	730	728							
TBIG	887	730	850						
TBIG1	897	899							
TBIGA	901	888							
TBIGB	908	910							
TBIGC	912	905							
TBIH	914	918							

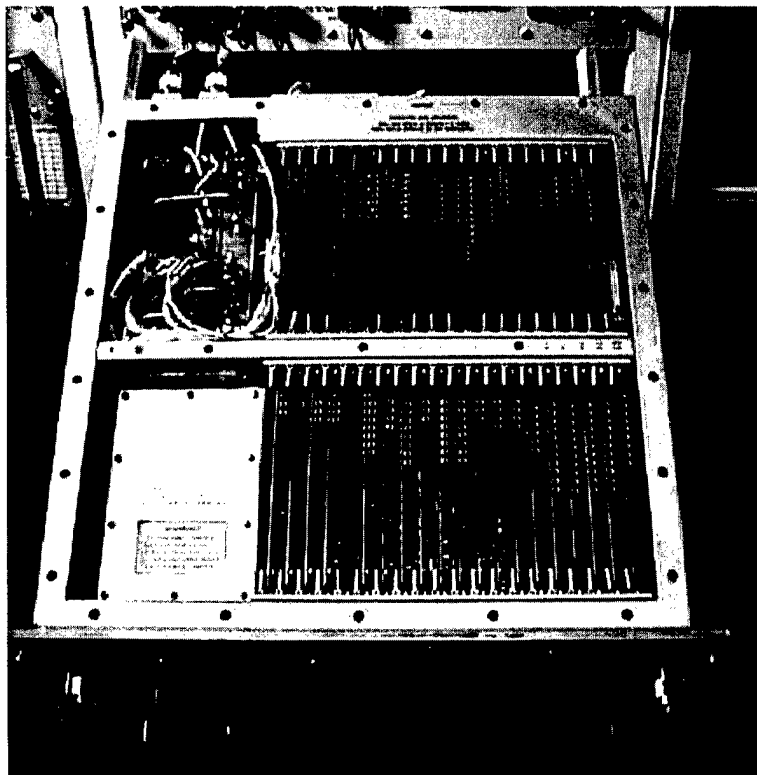
Label Name	Defined On	Referenced On							
TBIHA	921	942							
TBIHB	925	911							
TBIHC	928	926							
TBIJ	933	902							
TBIK	935	939							
TBIN	943	771	809	847	885				
TBIO	944								
TBIQ	949								
TBIQ1	952								
TBIR	962	958							
TBIRA	964	959	961						
TBIRB	977	970							
TBIS	979	951	963						
TDMA	156	64	172	183	3748	3777	3781	3791	3808
		3844	3854	4159					
TDMA1	185	159							
TDMAA	170	167							
TDMA B	171	169							
TDMAI	174	161							
TIBA	731	725							
TSTROM	28	24							

APPENDIX F

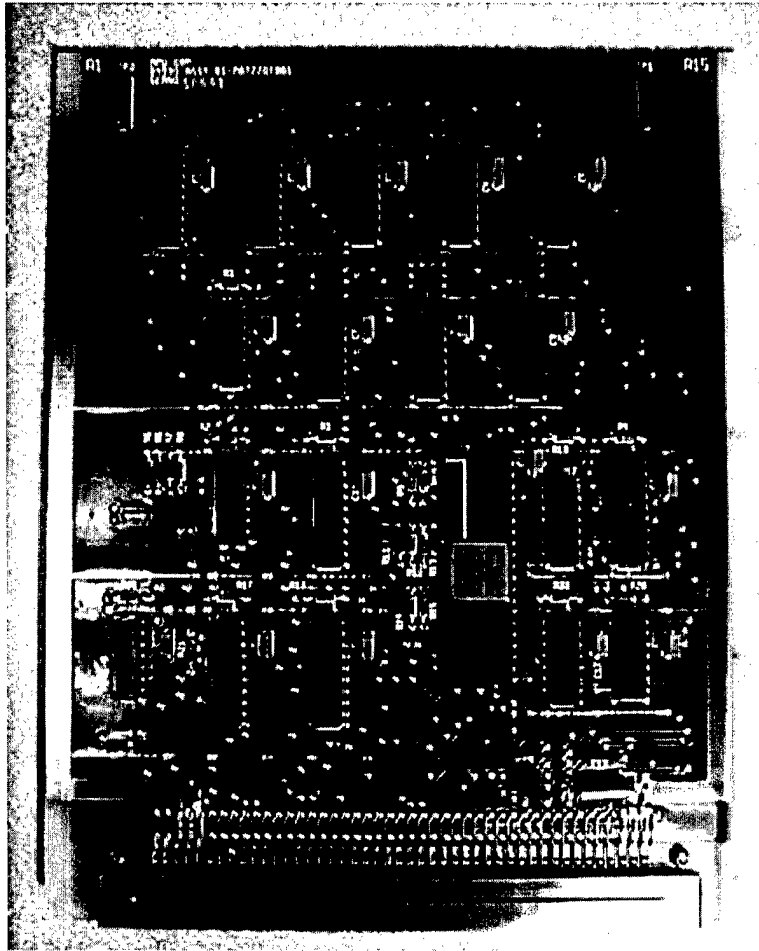
A. PHOTOS



TD-1271B/U Front Panel



TD-1271B/U Card Cage



DBP Processor Card

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center2
8725 John J. Kingman RD., Ste 0944
Fort Belvoir, VA 22060-6218

2. Dudley Knox Library2
Naval Postgraduate School
411 Dyer Road
Monterey, CA 93943-5150

3. Center for Naval Analysis.....1
4401 Ford Ave.
Alexandria, VA 22302-0268

4. Chief of Naval Research1
800 North Quincy St.
Arlington, VA 22217

5. Dr. Dan Boger, Chairman, Code CS1
Computer Science Dept.
Naval Postgraduate School
Monterey, CA 93943-5100

6. Dr. Man-Tak Shing, Code CS/SH.....1
Computer Science Dept.
Naval Postgraduate School
Monterey, CA 93943-5100

7. Kathleen M. Nelson1
SPAWAR SYSCEN SAN DIEGO
D844
San Diego, CA 92152-5150

8. Laurence Nixon.....1
SPAWAR SYSCEN SAN DIEGO
D832
San Diego, CA 92152-5150

9. Joe Hirschfelder1
SPAWAR SYSCEN SANDIEGO
D874
San Diego, CA 92152-5150